

Taller 1: Uso de Linux.

Investigación.

Para comprender mejor Linux, realice una pequeña búsqueda para responder las siguientes preguntas:

1. ¿Qué es una distribución Linux y que puede incluir?

Una distribución Linux es un conjunto de utilidades, software y diferentes builds del kernel de Linux. Existen muchas distribuciones de Linux, esto debido a que muchos individuos y compañías han creado sus propias versiones adaptadas a sus necesidades, y parten del proyecto OpenSource GNU Linux.

Componentes de una distribución Linux:

- **Kernel de Linux:** Cada distribución suele tener sus propios parches para el kernel de Linux y características que lo hacen cumplir con los objetivos de la distribución.
- **Utilidades del Core:** Incluye componentes del software como el bash, las utilidades de red, el sistema de archivos y otras piezas importantes que son la base de un sistema operativo.
- **Software suplementario:** Esto incluye las aplicaciones de productividad como las de escritorio, hojas de cálculo, procesadores de texto, los juegos, antivirus, etc.
- **Servicios del sistema (servicios de red, autenticación, servicios de seguridad):** En esta categoría se encuentran todos los servicios alrededor del sistema operativo, e incluyen entre otros los servicios de red, de autenticación y seguridad. Uno de los servicios más populares en las distribuciones es systemd. Otro es el manejador de paquetes, este se encarga de agregar, eliminar y actualizar software.

<https://apuntes.de/linux-certificacion-lpi/distribuciones-de-linux>

2. ¿Qué es un shell?

El Shell es una herramienta del sistema operativo que sirve de mediador entre el usuario y el núcleo del sistema operativo. Cada vez que realizas una acción sobre tu equipo, el shell está actuando sin que lo notes.

También llamado intérprete de comandos, este software provee de forma nativa un grupo de instrucciones con parámetros específicos que facilitan acciones sobre los distintos recursos a los que tiene acceso el sistema operativo.

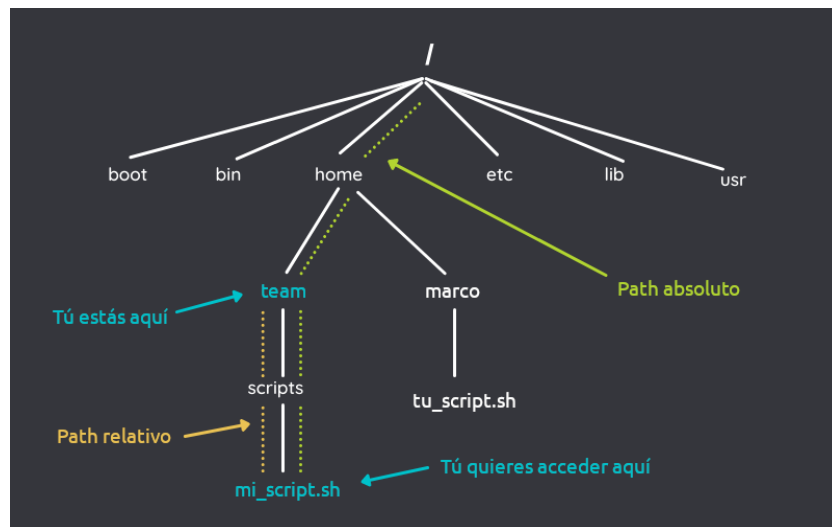
<https://www.dongee.com/tutoriales/que-es-shell/>

3. ¿Cuál es la diferencia entre una ruta relativa y una ruta absoluta a un archivo?

La ruta absoluta siempre comienza en el directorio raíz (/). Por ejemplo, **/home/itsfoss/scripts/mi_scripts.sh**.

Una ruta relativa comienza desde el directorio actual. Por ejemplo, si se encuentra en el directorio `/home` y desea acceder al archivo `mi_script.sh`, puede utilizar **itsfoss/scripts/mi_script.sh**.

En verde la ruta absoluta y en amarillo la ruta relativa.



<https://itsfoss.com/es/ruta-absoluta-relativa-linux/>

4. ¿Qué información brinda el comando **env**? ¿Cómo se relaciona con herramientas como Conda o Miniconda (¿son lo mismo?)

El comando **env**: Visualiza las variables de entorno y sus valores, pero el resultado no incluye las definiciones de funciones ni las variables definidas por el usuario.

<https://francisconi.org/linux/comandos/env>

Conda es un sistema de gestión de paquetes de código abierto y un sistema de gestión del entorno que se ejecuta en Windows, macOS y Linux. Conda instala, ejecuta y actualiza rápidamente los paquetes y sus dependencias y además también crea, guarda, carga y cambia fácilmente entre los entornos de su máquina local. Miniconda es una versión ligera de Conda.

<https://geoinnova.org/blog-territorio/las-facilidades-no-visibles-de-conda/>

El comando **env** se usa en Conda para verificar los entornos existentes en el equipo. Por defecto Conda crea un entorno que se llama `base`, y que contiene la versión base de

Python y los paquetes que necesita para funcionar todo bien. Entonces para ver los entornos que existen creados en el equipo, se ejecuta el siguiente comando:

```
conda env list
```

También se pueden exportar los entornos de Conda, crearlos y/o actualizarlos posteriormente.

Para exportar el entorno, primero hay que asegurarse de que estamos trabajando sobre el entorno que nos interese y escribimos la siguiente instrucción:

```
conda env export > environment.yml
```

Este comando lo que hace es exportar todos los datos relativos al entorno en el que estamos trabajando en un archivo .yml, un archivo de texto plano.

Además, si usamos el nombre environment.yml, conda reconocerá este nombre, lo que nos será de mucha ayuda al clonarlo.

Por ejemplo, podemos recrear exactamente ese entorno en otro nuevo con:

```
conda env create -f environment.yml
```

Pero si ya estamos dentro de una carpeta que tiene ese archivo, lo usará automáticamente para instalar las dependencias porque reconocerá el nombre del archivo. Pero de todas formas es mucho mejor si se lo indicamos.

También se pueden instalar en el entorno actual con:

```
conda env update -f .\environment.yml
```

<https://www.campusmvp.es/recursos/post/como-gestionar-diferentes-entornos-para-python-con-conda.aspx>

5. ¿Para qué son útiles los pipes de UNIX? Muestre un ejemplo de su utilidad.

Los pipes son una herramienta potente en Linux, el sistema operativo de código abierto. Linux Pipes permite procesar secuencialmente una serie de comandos referentes a un conjunto de datos, o mover eficazmente los datos de un lado a otro entre comandos, por ejemplo.

<https://www.ionos.es/digitalguide/servidores/configuracion/pipes-linux>

Ejemplo de utilidad.

Si queremos ver el contenido del directorio /usr/bin podemos utilizar el comando **ls** como sigue:

```
ls /usr/bin
```

El problema es que este directorio contiene demasiados ficheros y éstos desaparecen de la pantalla muy rápidamente al visualizarlos. Para solucionarlo, sería útil tener alguna forma de visualizar textos que dividiera la visualización en páginas a través de las cuales pudiéramos navegar con mayor comodidad.

El comando **more** realiza una función como la que hemos descrito. Así, si deseamos visualizar página a página el contenido de un fichero llamado **HOWTOUSE** con información sobre el uso del programa **cdda2wav** situado en el directorio **/usr/share/doc/cdda2wav**, podemos hacer lo siguiente:

```
more /usr/share/doc/zip-3.0/README
```

El comando visualiza el contenido del fichero pero deteniéndose cuando el texto ocupa toda la página e indicando con un texto en la parte inferior el porcentaje del total que se ha mostrado hasta el momento. Pulsando la barra espaciadora podemos ir viendo el resto de páginas y pulsando la tecla 'q' finalizar la visualización.

Aunque el comando **more** parece adecuado para el problema que planteamos antes, la cuestión es ¿cómo lo aplicamos al resultado de visualizar el contenido de un directorio con **ls** si necesita que le proporcionemos un fichero como argumento? La solución consiste en encadenar la salida del comando **ls** y usarla como entrada para el comando **more** a través de un pipe. Para ello utilizaremos el símbolo '|' de la siguiente forma:

```
ls /usr/bin | more
```

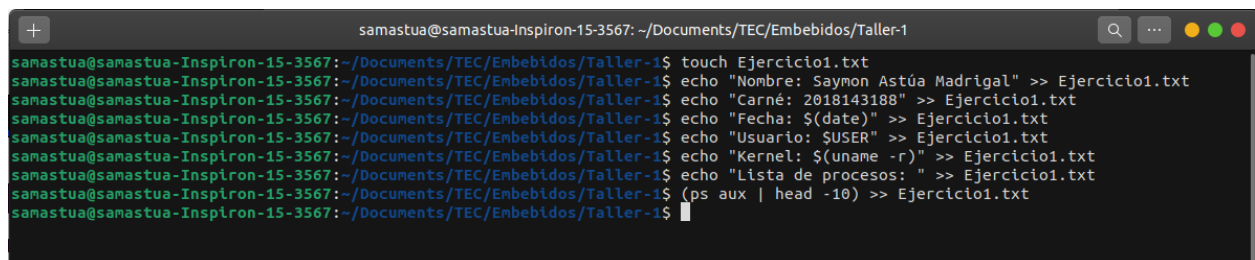
Como resultado, obtenemos la salida que proporciona **ls** pero dividida en páginas.

<http://multimedia.uoc.edu/blogs/linux/tuberias-pipes/>

Práctica.

Ejercicio 1.

Comandos necesarios.



```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ touch Ejercicio1.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ echo "Nombre: Saymon Astúa Madrigal" >> Ejercicio1.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ echo "Carné: 2018143188" >> Ejercicio1.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ echo "Fecha: $(date)" >> Ejercicio1.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ echo "Usuario: $USER" >> Ejercicio1.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ echo "Kernel: $(uname -r)" >> Ejercicio1.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ echo "Lista de procesos: " >> Ejercicio1.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ (ps aux | head -10) >> Ejercicio1.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$
```

Contenido del archivo Ejercicio1.txt

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ cat Ejercicio1.txt
Nombre: Saymon Astúa Madrigal
Carné: 2018143188
Fecha: Thu Feb 16 02:16:52 PM CST 2023
Usuario: samastua
Kernel: 5.15.0-60-generic
Lista de procesos:
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.4  0.1 168496 13940 ?        Ss   13:58   0:05 /sbin/init splash
root         2  0.0  0.0      0     0 ?        S    13:58   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        I<   13:58   0:00 [rcu_gp]
root         4  0.0  0.0      0     0 ?        I<   13:58   0:00 [rcu_par_gp]
root         5  0.0  0.0      0     0 ?        I<   13:58   0:00 [slub_flushwq]
root         6  0.0  0.0      0     0 ?        I<   13:58   0:00 [netns]
root         8  0.0  0.0      0     0 ?        I<   13:58   0:00 [kworker/0:0H-events_highpri]
root         9  0.1  0.0      0     0 ?        I    13:58   0:01 [kworker/u8:0-phy0]
root        10  0.0  0.0      0     0 ?        I<   13:58   0:00 [mm_percpu_wq]
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$
```

Ejercicio 2.

Generar un nuevo archivo (Ejercicio2.txt) que contenga la información de checksum MD5 del archivo Ejercicio1.txt.

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ mkdir Ej2
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ ls
Ej2
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ touch Ejercicio2.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ ls
Ej2 Ejercicio1.txt Ejercicio2.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ md5sum Ejercicio1.txt >> Ejercicio2.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ cat Ejercicio2.txt
9aec8efd03daa144fae13d5ea1dc7bbf Ejercicio1.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$
```

Realizar un archivamiento (Ejercicios.tar) a partir de ambos archivos (Ejercicio1.txt y Ejercicio2.txt).

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ tar -cvf Ejercicios.tar Ejercicio1.txt Ejercicio2.txt
Ejercicio1.txt
Ejercicio2.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ tree
├── Ej2
│   ├── Ejercicio1.txt
│   ├── Ejercicio2.txt
│   └── Ejercicios.tar
1 directory, 3 files
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$
```

Copiar el archivo Ejercicios.tar en el directorio recién creado Ej2.

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ cp Ejercicios.tar ./Ej2
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ tree
├── Ej2
│   ├── Ejercicios.tar
│   ├── Ejercicio1.txt
│   ├── Ejercicio2.txt
│   └── Ejercicios.tar
1 directory, 4 files
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$
```

Ingresar al directorio Ej2 y extraer allí el archivo Ejercicios.tar

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1/Ej2
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ cd Ej2/
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej2$ tar xfv Ejercicios.tar
Ejercicio1.txt
Ejercicio2.txt
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej2$ tree
├── Ejercicio1.txt
├── Ejercicio2.txt
└── Ejercicios.tar
0 directories, 3 files
```

Ejercicio 3.

Crear una imagen de archivo en blanco (empty.img), con un tamaño de 1KB.

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1/Ej3
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej2$ cd ..
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ mkdir Ej3
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1$ cd Ej3
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$ fallocate -l 1k empty.img
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$ ls -lh empty.img
-rw-rw-r-- 1 samastua samastua 1.0K Feb 16 14:32 empty.img
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$
```

Cambiar permisos de archivo para convertirlo en ejecutable (permiso de ejecución).

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1/Ej3
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$ chmod +x empty.img
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$ ls -lh empty.img
-rwxrwxr-x 1 samastua samastua 1.0K Feb 16 14:32 empty.img
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$
```

Escribir (por comandos) al archivo empty.img las siguientes líneas (5 líneas en total):

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1/Ej3
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$ cat <<EOF > empty.img
> clear
> wget http://www.bolis.com/onyx/random/stuff/sounds/murray/murrays.wav
> aplay murrays.wav
> mv murrays.wav .murrays.wav
> echo "Aquí no ha pasado nada"
> EOF
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$
```

Renombrar el archivo empty.img a script.x

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1/Ej3
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$ mv empty.img script.x
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$ ls
script.x
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej3$
```

Ejecutar script.x (¿Qué debe suceder en este punto?).

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1/Ej3
--2023-02-16 14:50:34-- http://www.bolis.com/onyx/random/stuff/sounds/murray/murrays.wav
Resolving www.bolis.com (www.bolis.com)... 173.236.153.129
Connecting to www.bolis.com (www.bolis.com)|173.236.153.129|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 113452 (111K) [audio/x-wav]
Saving to: 'murrays.wav'

murrays.wav                               100%[=====] 110.79K  473KB/s   in 0.2s

2023-02-16 14:50:35 (473 KB/s) - 'murrays.wav' saved [113452/113452]

Playing WAVE 'murrays.wav' : Signed 16 bit Little Endian, Rate 22050 Hz, Mono
Aquí no ha pasado nada
```

Lo que sucede es que se ejecutan las 5 líneas dentro del archivo. En resumen lo que hace es descargar un sonido y reproducirlo.

Ejercicio 4.

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1/Ej4
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej4$ wget https://archive.org/stream/laodisea00homeoft/laodisea00homeoft_djvu.txt | grep -c "Zeus"
--2023-02-16 15:00:52-- https://archive.org/stream/laodisea00homeoft/laodisea00homeoft_djvu.txt
Resolving archive.org (archive.org)... 207.241.224.2
Connecting to archive.org (archive.org)|207.241.224.2|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'laodisea00homeoft_djvu.txt'

laodisea00homeoft_djvu.txt                [ <=> ] 846.95K  1.39MB/s   in 0.6s

2023-02-16 15:00:53 (1.39 MB/s) - 'laodisea00homeoft_djvu.txt' saved [867277]

0
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej4$ ls
laodisea00homeoft_djvu.txt  laodisea00homeoft_djvu.txt.1
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej4$ less laodisea00homeoft_djvu.txt | grep -c "Zeus"
256
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej4$
```

Ejercicio 5.

```
samastua@samastua-Inspiron-15-3567: ~/Documents/TEC/Embebidos/Taller-1/Ej5
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej5$ usb-devices

T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=480 MxCh=12
D: Ver= 2.00 Cls=09(hub ) Sub=00 Prot=01 MxPS=64 #Cfgs= 1
P: Vendor=1d6b ProdID=0002 Rev=05.15
S: Manufacturer=Linux 5.15.0-60-generic xhci-hcd
S: Product=xHCI Host Controller
S: SerialNumber=0000:00:14.0
C: #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 4 IvL=256ms

T: Bus=01 Lev=01 Prnt=01 Port=02 Cnt=01 Dev#= 6 Spd=12 MxCh= 0
D: Ver= 2.00 Cls=00(>ifc ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1
P: Vendor=1532 ProdID=008c Rev=02.00
S: Manufacturer=Razer
S: Product=Razer DeathAdder V2 Mini
C: #Ifs= 3 Cfg#= 1 Atr=a0 MxPwr=500mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=03(HID ) Sub=01 Prot=02 Driver=usbhid
E: Ad=81(I) Atr=03(Int.) MxPS= 8 IvL=1ms
I: If#= 1 Alt= 0 #EPs= 1 Cls=03(HID ) Sub=00 Prot=01 Driver=usbhid
E: Ad=82(I) Atr=03(Int.) MxPS= 16 IvL=1ms
I: If#= 2 Alt= 0 #EPs= 1 Cls=03(HID ) Sub=00 Prot=01 Driver=usbhid
E: Ad=83(I) Atr=03(Int.) MxPS= 8 IvL=1ms

T: Bus=02 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=5000 MxCh= 6
D: Ver= 3.00 Cls=09(hub ) Sub=00 Prot=03 MxPS= 9 #Cfgs= 1
P: Vendor=1d6b ProdID=0003 Rev=05.15
S: Manufacturer=Linux 5.15.0-60-generic xhci-hcd
S: Product=xHCI Host Controller
S: SerialNumber=0000:00:14.0
C: #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr=0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 4 IvL=256ms
samastua@samastua-Inspiron-15-3567:~/Documents/TEC/Embebidos/Taller-1/Ej5$
```