



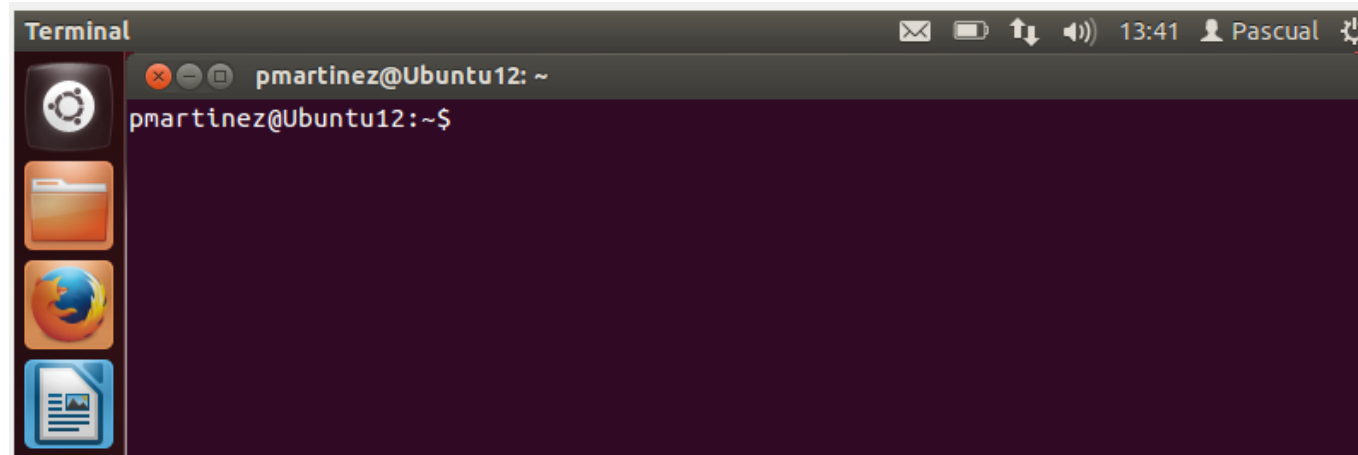
1º DAM/DAW Sistemas Informáticos

U2. Comandos en Linux.

Comandos. Estructura directorios

Interfaz de línea de comandos (CLI) en Linux. Shell

- La interfaz de línea de comandos, Shell, es un programa que recibe lo que se está escribiendo en un terminal y lo convierte en instrucciones para el sistema operativo. Permite acceder de ese modo a los servicios de un sistema operativo.
- Cuando el usuario escribe un comando, el intérprete, Shell, ejecuta esta orden. Con ese fin, utilizaremos el Shell de Ubuntu. Lo podemos encontrar escribiendo “terminal” en el inicio o bien, desde las Aplicaciones.



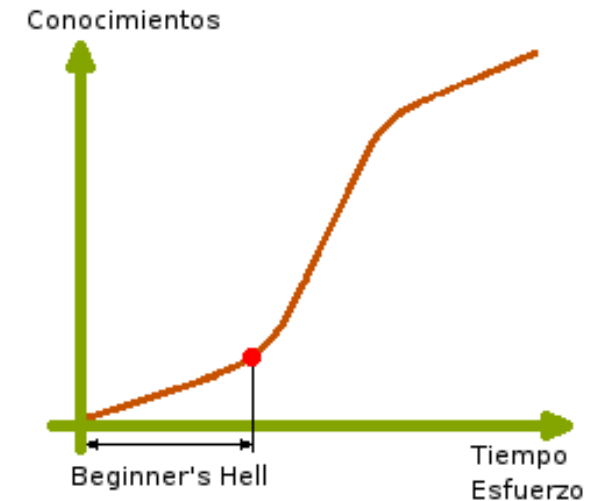
Uso de Shell. Ventajas e inconvenientes

- **Ventajas:**

- **Consume pocos recursos:** manejamos una aplicación sencilla y texto.
- **Amplitud de posibilidades:** permite ejecutar y combinar multitud de operaciones con diferentes utilidades.
- Posibilitan el desarrollo de **habilidades técnicas**, aplicables en múltiples escenarios.
- **Recomiendan su uso** un elevado porcentaje de expertos.

- **Inconvenientes:**

- **Complejidad de uso:** la interfaz es poco intuitiva, los comandos requieren preparación y consulta para su uso. En general la curva de aprendizaje es alta.



Curva de aprendizaje Linux

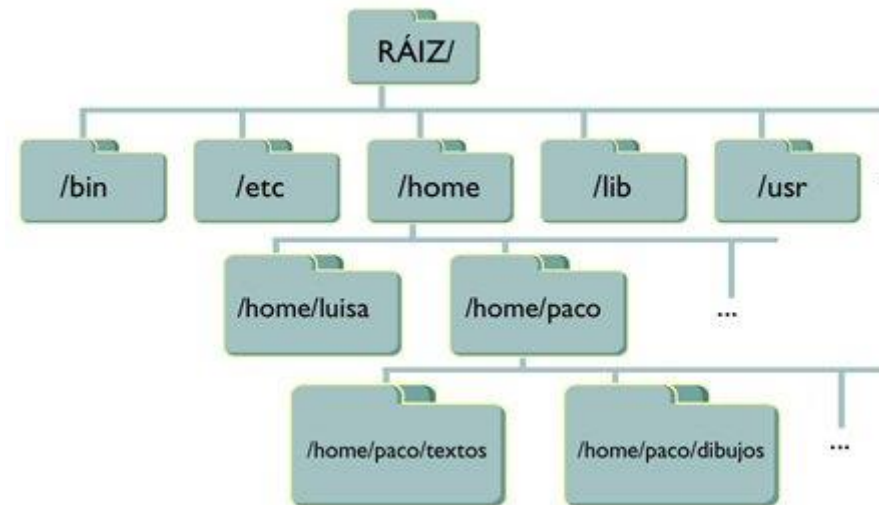
Shell. Algunos conceptos...

- **Bash:** por sus siglas en inglés, **B**ourne-**a**gain **S**hell. Es una aplicación de tipo Shell, muy extendida en su uso. Se desarrolló inicialmente para el S.O. Unix, actualmente viene como Shell en multitud de distribuciones Linux. Digamos que Shell es la denominación genérica de un tipo de herramienta y Bash es el nombre de una herramienta concreta de tipo Shell.
- **Prompt:** carácter o conjunto de caracteres que se muestran en una línea de comandos para indicar que está a la espera de órdenes.



Estructura de directorios. Rutas

- **Estructura de directorios:** se organiza como una estructura jerárquica de tipo árbol, donde el más alto nivel es el directorio raíz ("/") y todo lo demás se encuentra bajo él.
- **Ruta:** en el ámbito de la informática, una ruta (path, en inglés) es la forma de referenciar un archivo informático o directorio en un sistema de archivos de un sistema operativo.



Rutas. Absolutas y relativas

- Existen 2 modos de expresar las rutas:
 - **Rutas absolutas:** señalan la ubicación de un archivo o directorio desde el directorio raíz del sistema de archivos.
 - **Rutas relativas:** señalan la ubicación de un archivo o directorio desde nuestra posición actual en el sistema de archivos.
 - **Ejemplo:** suponemos que una persona tiene una imagen en su carpeta personal en Linux, ¿cómo podemos expresarlo?:
 - La ruta **absoluta** sería: **/home/usuario/imagen.jpg**
 - La ruta **relativa** sería:
 - Si estoy ubicado en el directorio raíz **"/**: en esta ubicación es **la misma** que la ruta absoluta , por definición.
 - Si estoy ubicado en el directorio **"/home/"**: **usuario/imagen.jpg**
 - Si estoy ubicado en **"/home/usuario/"**: **imagen.jpg**

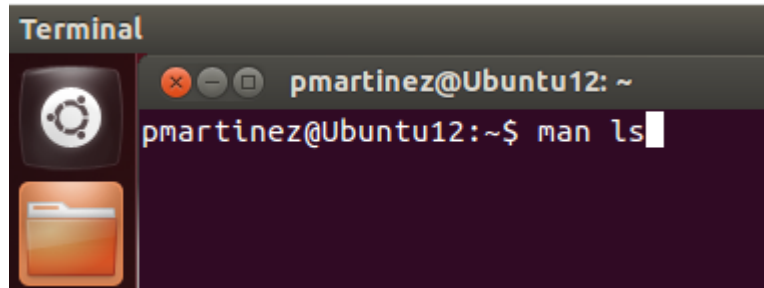
Comandos

- Un comando es una instrucción que se envía al sistema operativo, a través de una interfaz de línea de comandos.
- Tienen la siguiente sintaxis: **\$ nombre_comando arg1 arg2 ... argn**
- Por tanto, en la línea de comandos escribiremos el nombre del comando, seguido de uno o varios argumentos según las circunstancias.

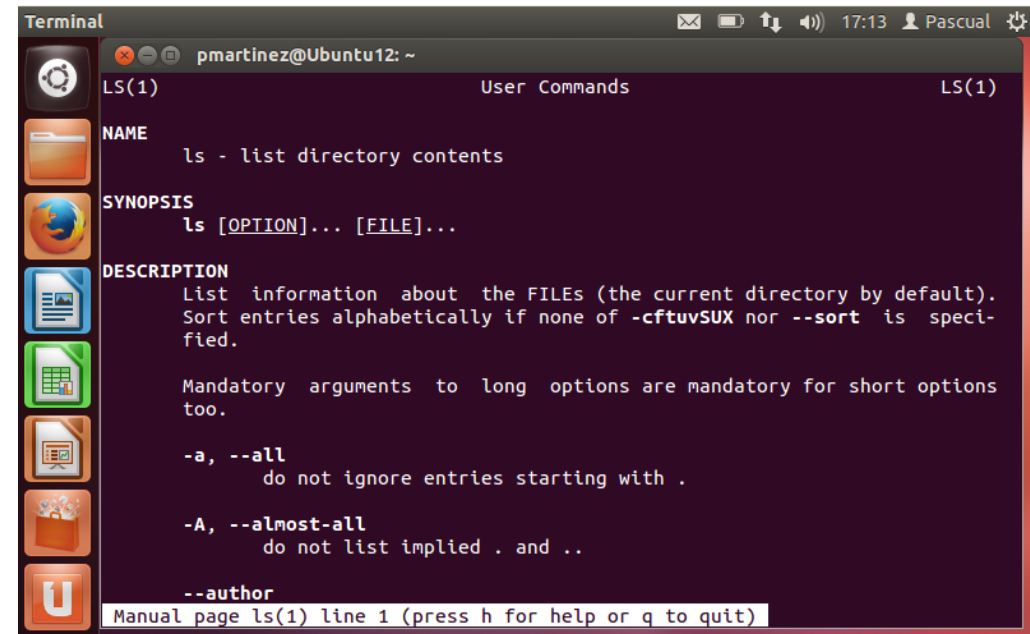


Comandos de ayuda

- **Comando man:** nos muestra el manual del comando que le pasemos como parámetro. Este comando nos servirá para poder consultar el funcionamiento del resto de comandos desde la interfaz de línea de comandos.



```
Terminal
pmartinez@Ubuntu12: ~
pmartinez@Ubuntu12:~$ man ls
```



```
Terminal
pmartinez@Ubuntu12: ~
LS(1) User Commands LS(1)

NAME
ls - list directory contents

SYNOPSIS
ls [OPTION]... [FILE]...

DESCRIPTION
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
fied.

Mandatory arguments to long options are mandatory for short options
too.

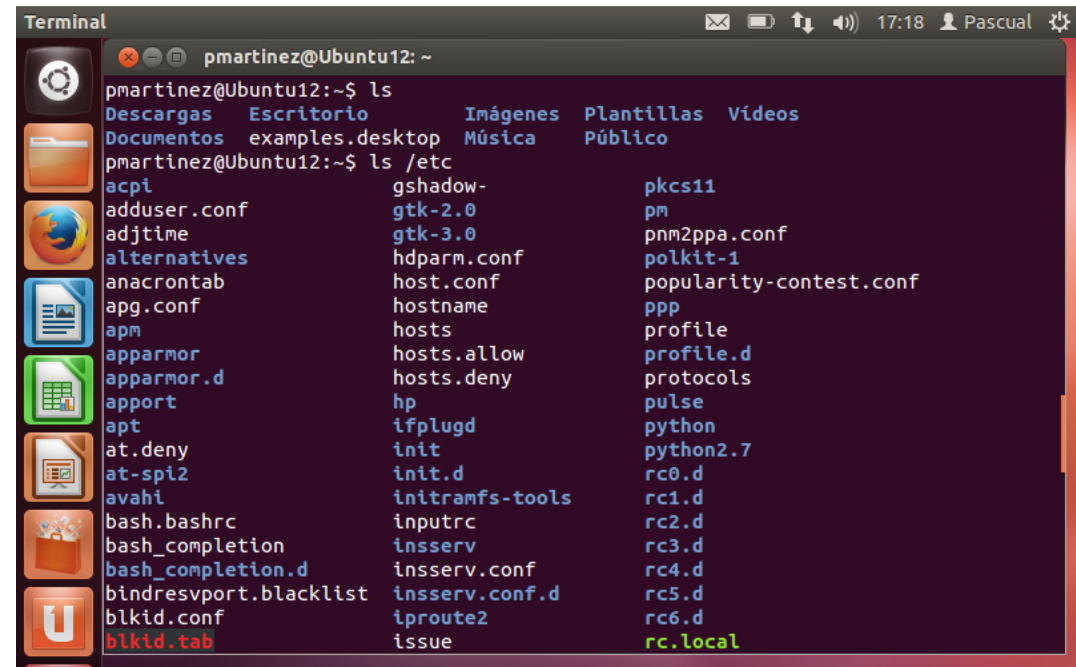
-a, --all
do not ignore entries starting with .

-A, --almost-all
do not list implied . and ..

--author
Manual page ls(1) line 1 (press h for help or q to quit)
```


Comandos básicos

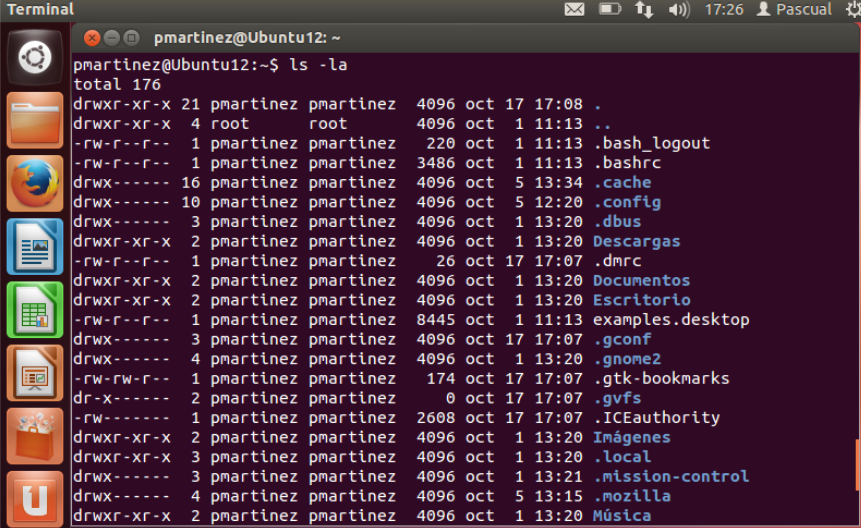
- **Comando ls:** muestra un listado con los ficheros y directorios de un directorio o carpeta determinada. Si no indicamos parámetros, muestra el contenido del directorio actual. Podemos pasarle una ruta como parámetro y nos muestra su contenido.



```
Terminal
pmartinez@Ubuntu12: ~
pmartinez@Ubuntu12:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  Videos
Documentos examples.desktop Música      Público
pmartinez@Ubuntu12:~$ ls /etc
acpi          gshadow-          pkcs11
adduser.conf  gtk-2.0            pm
adjtime       gtk-3.0            pnm2ppa.conf
alternatives  hdparm.conf        polkit-1
anacrontab    host.conf          popularity-contest.conf
apg.conf      hostname           ppp
apm           hosts              profile
apparmor      hosts.allow        profile.d
apparmor.d    hosts.deny          protocols
appport       hp                  pulse
apt           ifplugd             python
at.deny       init                python2.7
at-spi2       init.d              rc0.d
avahi         initramfs-tools     rc1.d
bash.bashrc   inputrc             rc2.d
bash_completion insserv             rc3.d
bash_completion.d insserv.conf        rc4.d
bindresvport.blacklist insserv.conf.d      rc5.d
blkid.conf    iproute2            rc6.d
blkid.tab     issue               rc.local
```

Comandos básicos

- **Comando ls:**
 - Con el **parámetro -l**, nos mostrará información más detallada.
 - Con el **parámetro -a**, nos mostrará los elementos ocultos del sistema.
 - Con el **parámetro ***, nos mostrará un listado recursivo; es decir, mostrará el contenido de las carpetas contenidas en el directorio listado.
 - Se pueden combinar los parámetros.



```
Terminal
pmartinez@Ubuntu12: ~
pmartinez@Ubuntu12:~$ ls -la
total 176
drwxr-xr-x 21 pmartinez pmartinez 4096 oct 17 17:08 .
drwxr-xr-x  4 root      root      4096 oct  1 11:13 ..
-rw-r--r--  1 pmartinez pmartinez  220 oct  1 11:13 .bash_logout
-rw-r--r--  1 pmartinez pmartinez 3486 oct  1 11:13 .bashrc
drwx----- 16 pmartinez pmartinez 4096 oct  5 13:34 .cache
drwx----- 10 pmartinez pmartinez 4096 oct  5 12:20 .config
drwx-----  3 pmartinez pmartinez 4096 oct  1 13:20 .dbus
drwxr-xr-x  2 pmartinez pmartinez 4096 oct  1 13:20 Descargas
-rw-r--r--  1 pmartinez pmartinez   26 oct 17 17:07 .dmrc
drwxr-xr-x  2 pmartinez pmartinez 4096 oct  1 13:20 Documentos
drwxr-xr-x  2 pmartinez pmartinez 4096 oct  1 13:20 Escritorio
-rw-r--r--  1 pmartinez pmartinez 8445 oct  1 11:13 examples.desktop
drwx-----  3 pmartinez pmartinez 4096 oct 17 17:07 .gconf
drwx-----  4 pmartinez pmartinez 4096 oct  1 13:20 .gnome2
-rw-rw-r--  1 pmartinez pmartinez  174 oct 17 17:07 .gtk-bookmarks
dr-x-----  2 pmartinez pmartinez    0 oct 17 17:07 .gvfs
-rw-----  1 pmartinez pmartinez 2608 oct 17 17:07 .ICEauthority
drwxr-xr-x  2 pmartinez pmartinez 4096 oct  1 13:20 Imágenes
drwxr-xr-x  3 pmartinez pmartinez 4096 oct  1 13:20 .local
drwx-----  3 pmartinez pmartinez 4096 oct  1 13:21 .mission-control
drwx-----  4 pmartinez pmartinez 4096 oct  5 13:15 .mozilla
drwxr-xr-x  2 pmartinez pmartinez 4096 oct  1 13:20 Música
```

Comandos básicos

- **Caracteres comodín:**

- Podemos utilizar el carácter `*` para sustituirlo por cualquier cadena de 0 o más caracteres. Por ejemplo, con el siguiente comando podríamos listar ficheros que empiecen por `h` y tengan a continuación 0 o más caracteres:

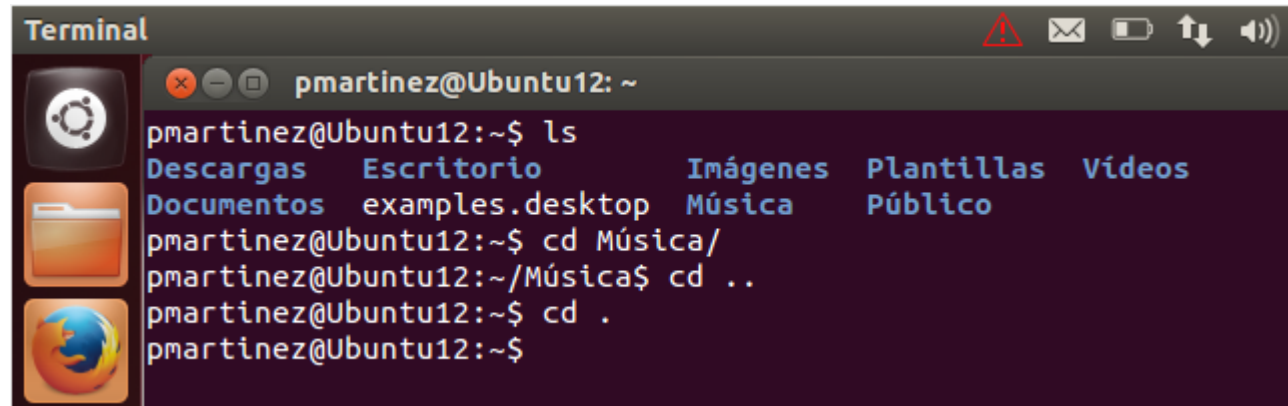
`ls h*`

- Podemos utilizar el carácter `?` para sustituirlo por cualquier carácter individual. Por ejemplo, con el siguiente comando podríamos listar ficheros que empiecen por `h` y tengan exactamente 5 caracteres más:

`ls h?????`

Comandos básicos

- **Comando cd:** para desplazarnos a un directorio, utilizaremos la sintaxis **cd nombre_directorio**
 - Por ejemplo:
 - Situado en el directorio raíz (/home/usuario), prueba a moverte al directorio Música.
 - Para volver al directorio anterior, simplemente hay que hacer **cd ..**
 - Si utilizas **cd .** te quedas en la carpeta actual.

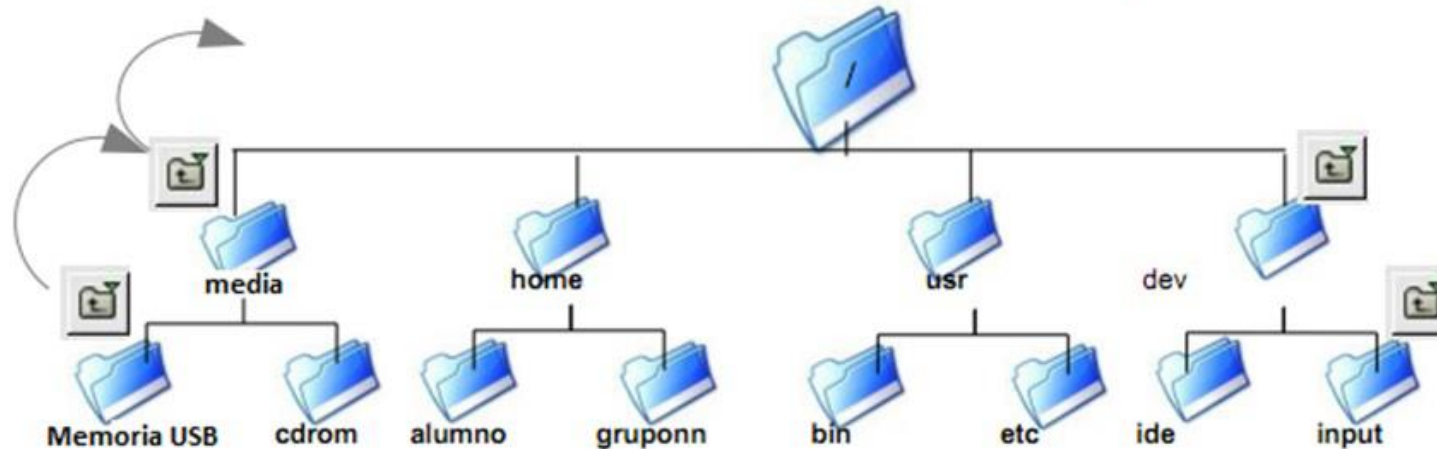


```
Terminal
pmartinez@Ubuntu12: ~
pmartinez@Ubuntu12:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  Vídeos
Documentos examples.desktop  Música    Público
pmartinez@Ubuntu12:~$ cd Música/
pmartinez@Ubuntu12:~/Música$ cd ..
pmartinez@Ubuntu12:~$ cd .
pmartinez@Ubuntu12:~$
```

Comandos básicos

- **Comando cd:** Imagina que tenemos una estructura de directorios como la de la imagen. Si estamos situados en el directorio `"/media/Memoria USB"`, con el comando **cd ..** nos situaríamos en `"/media"`, y si volvemos a hacer un **cd ..** nos situaríamos en `"/`". Si desde aquí hacemos un **cd home/alumno** nos situaríamos en el directorio `"/home/alumno"`. Además, también podemos hacer todo este movimiento de una sola vez con **cd ../../home/alumno**, e incluso acceder al directorio apuntando directamente a su ruta a partir del directorio raíz `"/`", con **cd /home/alumno**.

Árbol de directorios de un disco duro con sistema operativo Linux



Comandos básicos

- **Comandos mkdir rmdir:**

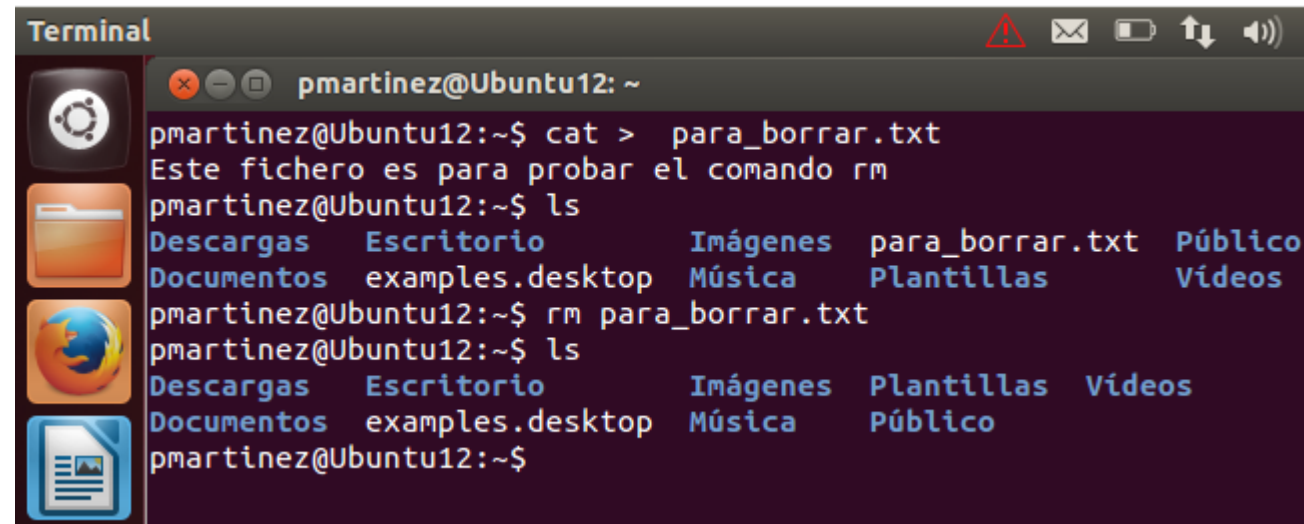
- El comando **mkdir** nos permite crear un directorio vacío, para ello ejecutaremos este comando seguido del nombre del directorio que deseemos. Si hacemos un **ls** posteriormente, podemos ver que hemos creado ese directorio.
- El comando **rmdir** nos permite borrar un directorio vacío, para ello ejecutaremos este comando seguido del nombre del directorio que deseemos. Si hacemos un **ls** podemos ver que el directorio ya no existe una vez borrado.



```
Terminal
pmartinez@Ubuntu12: ~
pmartinez@Ubuntu12:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  Videos
Documentos examples.desktop  Música    Público
pmartinez@Ubuntu12:~$ mkdir Directorio_nuevo
pmartinez@Ubuntu12:~$ ls
Descargas  Documentos  examples.desktop  Música  Público
Directorio_nuevo  Escritorio  Imágenes  Plantillas  Videos
pmartinez@Ubuntu12:~$ rmdir Directorio_nuevo/
pmartinez@Ubuntu12:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  Videos
Documentos examples.desktop  Música    Público
pmartinez@Ubuntu12:~$
```

Comandos básicos

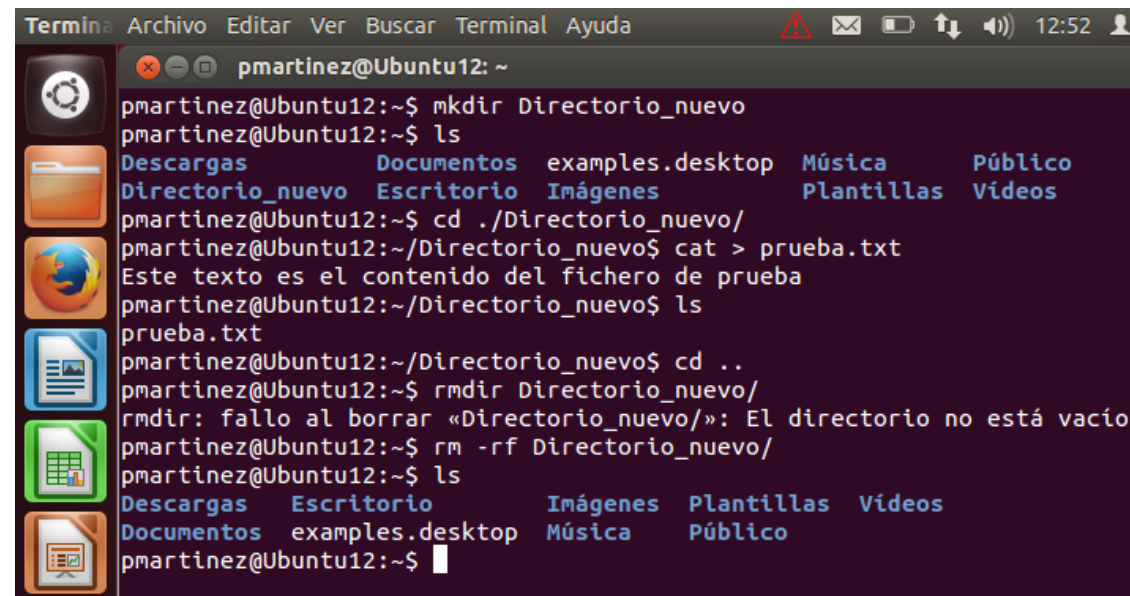
- **Comando rm:** nos permite borrar un fichero, para ello ejecutaremos este comando seguido del nombre del directorio que deseemos. Si hacemos un **rm** de un fichero de prueba que hubiéramos creado (para ello puedes usar el comando touch, o bien *en la imagen verás que yo he creado el fichero mediante el comando “cat > nombre_fichero” + “intro” + “texto_del fichero” + “Ctrl+D”*) podremos ver que ya no existe.



```
Terminal
pmartinez@Ubuntu12: ~
pmartinez@Ubuntu12:~$ cat > para_borrar.txt
Este fichero es para probar el comando rm
pmartinez@Ubuntu12:~$ ls
Descargas  Escritorio  Imágenes  para_borrar.txt  Público
Documentos examples.desktop  Música    Plantillas  Vídeos
pmartinez@Ubuntu12:~$ rm para_borrar.txt
pmartinez@Ubuntu12:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  Vídeos
Documentos examples.desktop  Música    Público
pmartinez@Ubuntu12:~$
```

Comandos básicos

- **Comando rm:** para borrar varios ficheros de un directorio, podemos utilizar el carácter comodín * . Si intentamos borrar un directorio, puede que no nos deje porque tiene subdirectorios o ficheros dentro. Para solucionar esto, podemos borrar el contenido del directorio previamente, o bien si queremos borrar directorios y subdirectorios y además forzar el borrado, utilizaremos los parámetros **-rf** .



```
Terminal Archivo Editar Ver Buscar Terminal Ayuda 12:52
pmartinez@Ubuntu12: ~
pmartinez@Ubuntu12:~$ mkdir Directorio_nuevo
pmartinez@Ubuntu12:~$ ls
Descargas  Documentos  examples.desktop  Música  Público
Directorio_nuevo  Escritorio  Imágenes  Plantillas  Vídeos
pmartinez@Ubuntu12:~$ cd ./Directorio_nuevo/
pmartinez@Ubuntu12:~/Directorio_nuevo$ cat > prueba.txt
Este texto es el contenido del fichero de prueba
pmartinez@Ubuntu12:~/Directorio_nuevo$ ls
prueba.txt
pmartinez@Ubuntu12:~/Directorio_nuevo$ cd ..
pmartinez@Ubuntu12:~$ rmdir Directorio_nuevo/
rmdir: fallo al borrar «Directorio_nuevo/»: El directorio no está vacío
pmartinez@Ubuntu12:~$ rm -rf Directorio_nuevo/
pmartinez@Ubuntu12:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  Vídeos
Documentos  examples.desktop  Música  Público
pmartinez@Ubuntu12:~$
```


Comandos básicos

- **Comandos cp y mv:**

- El comando **cp** se utiliza para copiar un fichero. La sintaxis es:
cp fichero_origen fichero_destino. Para ello, el fichero origen debe existir.
- Si queremos copiar varios ficheros origen a la vez a un mismo directorio destino, podemos emplear la sintaxis: **cp origen origen origen destino**.
- Si queremos copiar todos los ficheros de un directorio añadiremos el carácter ***** para indicar que queremos copiar todos los ficheros de ese directorio.
- Si además queremos copiar los subdirectorios de manera recurrente, añadiremos el parámetro **-r**.
- El comando **mv** se emplea para mover directorios y ficheros. La sintaxis es muy similar a la del comando cp, pero el resultado es que se elimina cada origen una vez copiado en destino.

Comandos básicos

- Comandos cp y mv:

Ejemplos sobre lo indicado en la página anterior.

```

pmartinez@Ubuntu12: ~/borrar

pmartinez@Ubuntu12:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  Videos
Documentos examples.desktop  Música  Público
pmartinez@Ubuntu12:~$ mkdir Directorio_prueba
pmartinez@Ubuntu12:~$ cd Directorio_prueba/
pmartinez@Ubuntu12:~/Directorio_prueba$ cat > prueba.txt
Texto del fichero de prueba
pmartinez@Ubuntu12:~/Directorio_prueba$ cp prueba.txt copia_prueba.txt
pmartinez@Ubuntu12:~/Directorio_prueba$ cp prueba.txt otra_copia_prueba.txt
pmartinez@Ubuntu12:~/Directorio_prueba$ cp prueba.txt copia_prueba.txt Otro_Directorio/
pmartinez@Ubuntu12:~/Directorio_prueba$ cd Otro_Directorio/
pmartinez@Ubuntu12:~/Directorio_prueba/Otro_Directorio$ ls
copia_prueba.txt  prueba.txt
pmartinez@Ubuntu12:~/Directorio_prueba/Otro_Directorio$ rm *.*
pmartinez@Ubuntu12:~/Directorio_prueba/Otro_Directorio$ cd ..
pmartinez@Ubuntu12:~/Directorio_prueba$ cp *.* Otro_Directorio/
pmartinez@Ubuntu12:~/Directorio_prueba$ ls
copia_prueba.txt  otra_copia_prueba.txt  Otro_Directorio  prueba.txt
pmartinez@Ubuntu12:~/Directorio_prueba$ cd Otro_Directorio/
pmartinez@Ubuntu12:~/Directorio_prueba/Otro_Directorio$ ls
copia_prueba.txt  otra_copia_prueba.txt  prueba.txt
pmartinez@Ubuntu12:~/Directorio_prueba/Otro_Directorio$ cd ../../
pmartinez@Ubuntu12:~$ mkdir Directorio_prueba/Directorio_final
pmartinez@Ubuntu12:~$ cp -r Directorio_prueba/Otro_Directorio/ Directorio_prueba/Directorio_final/
pmartinez@Ubuntu12:~$ ls Directorio_prueba/Directorio_final/
Otro_Directorio
pmartinez@Ubuntu12:~$ ls Directorio_prueba/Directorio_final/Otro_Directorio/
copia_prueba.txt  otra_copia_prueba.txt  prueba.txt
pmartinez@Ubuntu12:~$ mkdir borrar
pmartinez@Ubuntu12:~$ mv Directorio_prueba/* borrar
pmartinez@Ubuntu12:~$ ls
borrar  Directorio_prueba  Escritorio  Imágenes  Plantillas  Videos
Descargas  Documentos  examples.desktop  Música  Público
pmartinez@Ubuntu12:~$ cd Directorio_prueba/
pmartinez@Ubuntu12:~/Directorio_prueba$ ls
pmartinez@Ubuntu12:~/Directorio_prueba$ cd ../borrar/
pmartinez@Ubuntu12:~/borrar$ ls
copia_prueba.txt  Directorio_final  otra_copia_prueba.txt  Otro_Directorio  prueba.txt
pmartinez@Ubuntu12:~/borrar$

```