

CENTRO NACIONAL DE SEGURIDAD DIGITAL



The background of the slide is a computer monitor displaying a Linux desktop. In the center of the screen is the Tux penguin mascot. The desktop has a dark theme with a grid of icons on the left and a panel at the bottom. The text 'Linux Essentials' is overlaid in the center of the screen.

Linux Essentials

Ing. Maurice Frayssinet Delgado
mfrayssinet@gmail.com
Tfl. (+51) 980.997.203

Unidad 2

Encontrar su camino en
un sistema Linux



Unidad 2 Encontrar su camino en un sistema Linux

- 2.1 Conceptos básicos de la línea de comandos.**
- 2.2 Uso de la línea de comandos para obtener ayuda.**
- 2.3 Uso de directorios y archivos de listas.**
- 2.4 Crear, mover y eliminar archivos.**

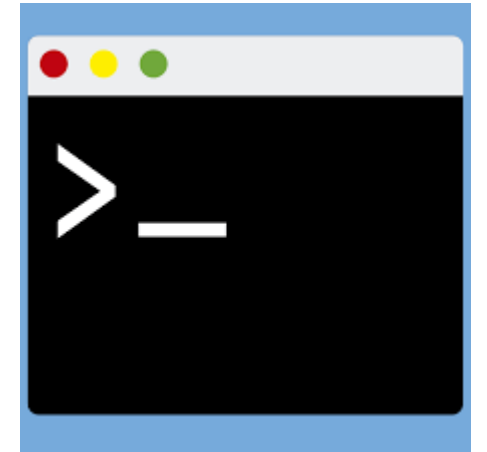




2.1 Conceptos básicos de la línea de comandos

Consola de comandos

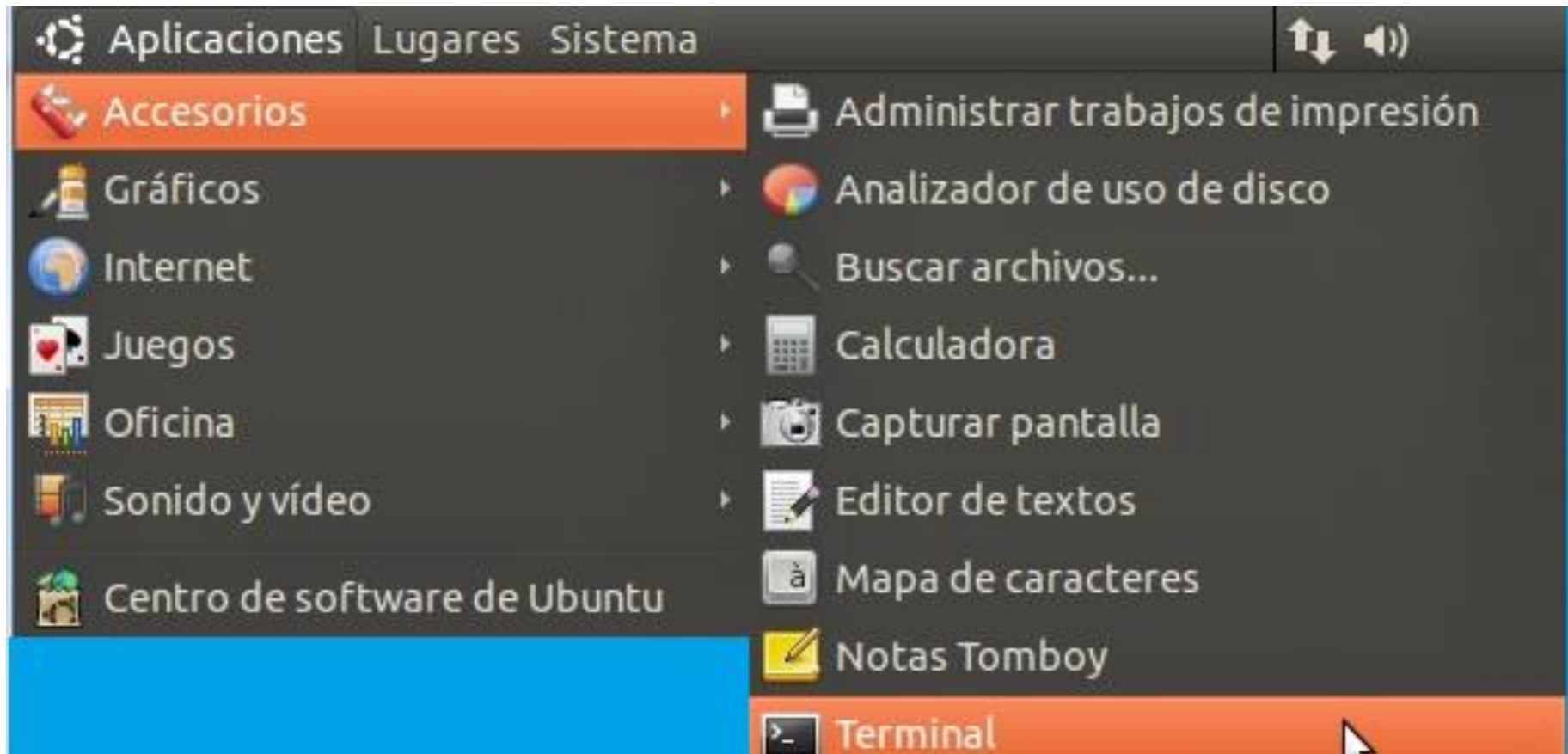
- Lo que ves cuando utilizas la mayor parte de los sistemas operativos es una representación gráfica de la estructura de los directorios y de las aplicaciones de tu ordenador.
- Esta representación no es más que una interfaz gráfica conocida como **GUI**, que significa **Graphical User Interface** «*Interfaz Gráfica de Usuario*».
- Esta interfaz está presente en todos los sistemas operativos más utilizados a nivel personal, ya sea en Linux, en macOS o en Windows.
- La consola de comandos sirve para poder ingresar los comandos para manejar el sistema operativo Linux



Ir a la consola de comandos



Ir a la consola de comandos



Ir a la consola de comandos

Presionan *Alt+F2*, escriben ***terminal*** o ***gnome-terminal*** y presionan ***enter***:



¿Qué es un comando?

- Un comando es un **programa de software** que al ejecutarlo en la **CLI (interfaz de línea de comandos)** realiza una acción en la computadora.
- Cuando se escribe un comando, el sistema operativo ejecuta un proceso que puede leer una entrada, manipular datos y producir la salida. Un comando ejecuta un proceso en el sistema operativo, y entonces causa que la computadora realice un trabajo.

Sintaxis comando Linux

Orden **opciones** **argumentos**

top

free **-m**

grep **-o** **manzana frutas.txt**

Algunos comandos importantes

- lscpu
- lspci
- lsusb
- Whoami
- uname
- History

Atajos Linux

- TAB: El tabulador permite autocompletar lo que se está escribiendo o navegar entre posibles opciones.
- CTRL + C: Sirve para matar un proceso que se esté ejecutando.
- CTRL + Z: Sirve para enviar al segundo plano el proceso que se está ejecutando. Si queremos volver al mismo, bastará con usar el comando fg.
- CTRL + D: Es el equivalente al comando exit, que sirve para cerrar la sesión de un usuario.
- CTRL + L: Este es el equivalente al comando clear y sirve para limpiar la pantalla de terminal. Es bastante útil, sobre todo cuando se ejecuta algún otro software de línea de comandos que no tiene un comando para limpiar pantalla y no se puede insertar el comando clear, como por ejemplo al utilizar el terminal de MySQL.
- CTRL + A: Mueve el cursor al inicio. Es útil si estamos escribiendo un comando largo y queremos corregir algo del inicio.

Atajos Linux

- CTRL + E: Mueve el cursor al final. De utilidad si queremos añadir algo al final del comando de forma rápida.
- CTRL + U: Con este atajo se borra la línea entera, independientemente de donde esté el cursor.
- CTRL + K: De esta forma conseguimos borrar desde la posición del cursor hasta el final de la línea.
- CTRL + W: Sirve para borrar la palabra que esté justo antes del cursor, o bien borra todas las letras desde la posición en la que está el mismo hasta el inicio de la palabra anterior.
- CTRL + Y: Permite deshacer el último borrado que se ha realizado.
- CTRL + R: Permite buscar dentro del historial de comandos que se han introducido. Es útil en caso de no recordar un comando que se ha introducido o uno que sea excesivamente largo, así se puede recuperar sin tener que volver a escribirlo. Cada vez que se pulsa CTRL + R la búsqueda va un paso más atrás.

Atajos Linux

- CTRL + G: Para salir de la búsqueda.
- CTRL + SHIFT + C: Sirve para copiar texto seleccionado en la terminal.
- CTRL + SHIFT + V: Permite pegar el texto copiado en la terminal.
- CTRL + S: De esta forma podemos pausar lo que se está imprimiendo por pantalla. Es útil para poder leer algo del texto que aparece durante una instalación, por ejemplo.
- CTRL + Q: Para reanudar la impresión por pantalla que se había pausado.

Apagar el sistema

Apagado predeterminado de Linux

- shutdown -h
- Shutdown

Reiniciar Linux de forma estándar

- shutdown -r

Apagar Linux inmediatamente

- shutdown -h 0
- shutdown now

Los comandos de apagado en Linux solo funcionan si tienes derechos de administrador. En algunos sistemas de Windows, este requisito se puede obviar anteponiendo el prefijo *sudo*: por ejemplo, en *sudo shutdown now* o *sudo shutdown r 0*.

Reiniciar el sistema

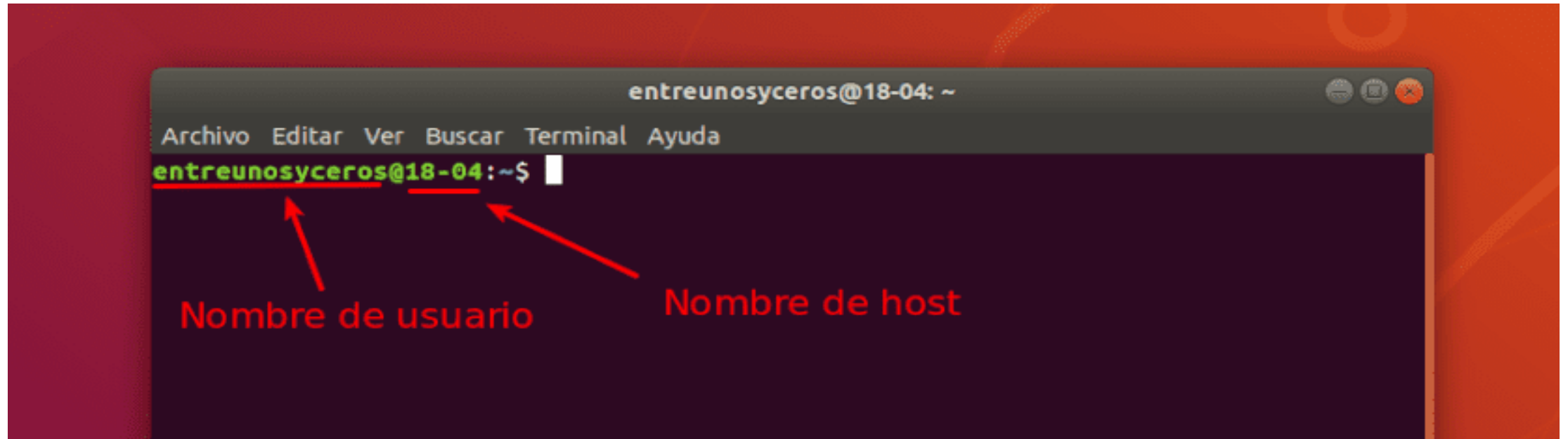
- `reboot`
- `init 6`
- `shutdown -r now`

Personaliza el Bash prompt

En BASH, podemos personalizar y cambiar el prompt en la forma que queramos. Solo habrá que **cambiar el valor de la variable de entorno *PS1***. En cada máquina se verá diferente nombre de usuario y nombre de host.

En el equipo en el que voy a probar estos ejemplos, entreunosyceros es mi nombre de usuario y 18-04 es mi nombre de host. A continuación vamos a **cambiar este prompt insertando algunos caracteres especiales llamados secuencias de escape**.

Personaliza el Bash prompt



Modificar 'nombreusuario@nombrehost' en el prompt

En este ejemplo vamos a reemplazar la parte de 'nombreusuario@nombrehost' por 'Hola@bienvenido>'.

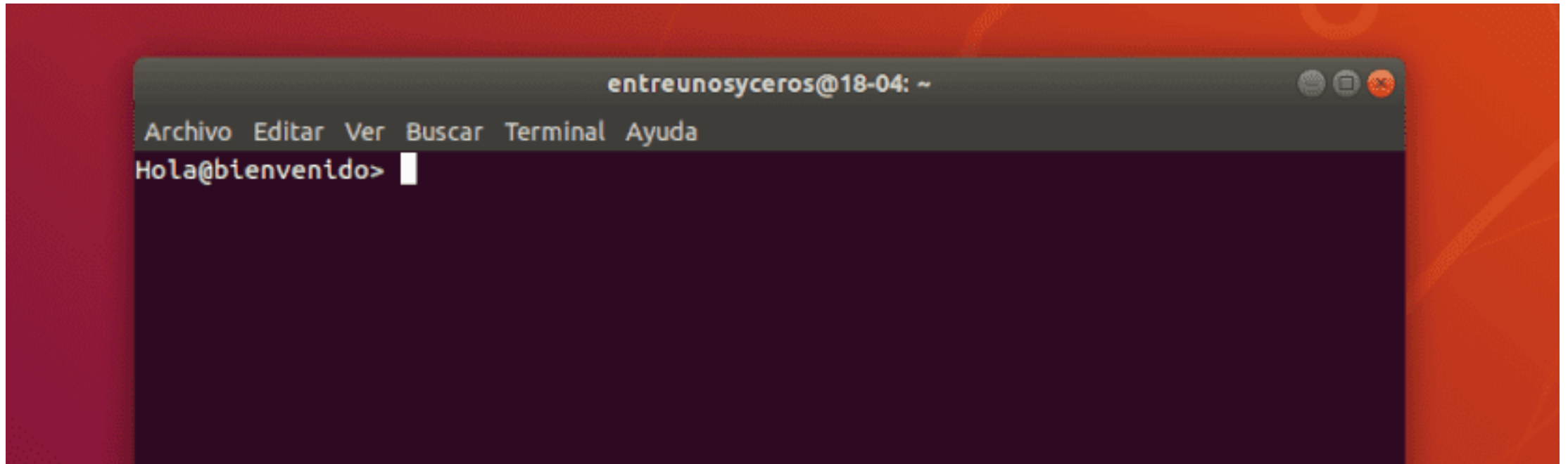
Para hacerlo, agrega lo siguiente a tu archivo `~/.bashrc`.

```
export PS1="Hola@bienvenido> "
```

Una vez terminado, guarda el archivo y vuelve a la terminal. **No olvides actualizar los cambios con el comando 'source ~/.bashrc'.**

Modificar 'nombreusuario@nombrehost' en el prompt

Aquí está la salida del prompt en mi Ubuntu 18.04 LTS.

A screenshot of a terminal window titled 'entreunosyceros@18-04: ~'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal output shows 'Hola@bienvenido>' followed by a cursor. The background of the terminal is dark purple, and the window title bar is dark gray.

```
entreunosyceros@18-04: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
Hola@bienvenido> |
```

Mostrar solo el nombre de usuario

Para mostrar solo el nombre de usuario, no hay más que cambiar la anterior línea por la siguiente:



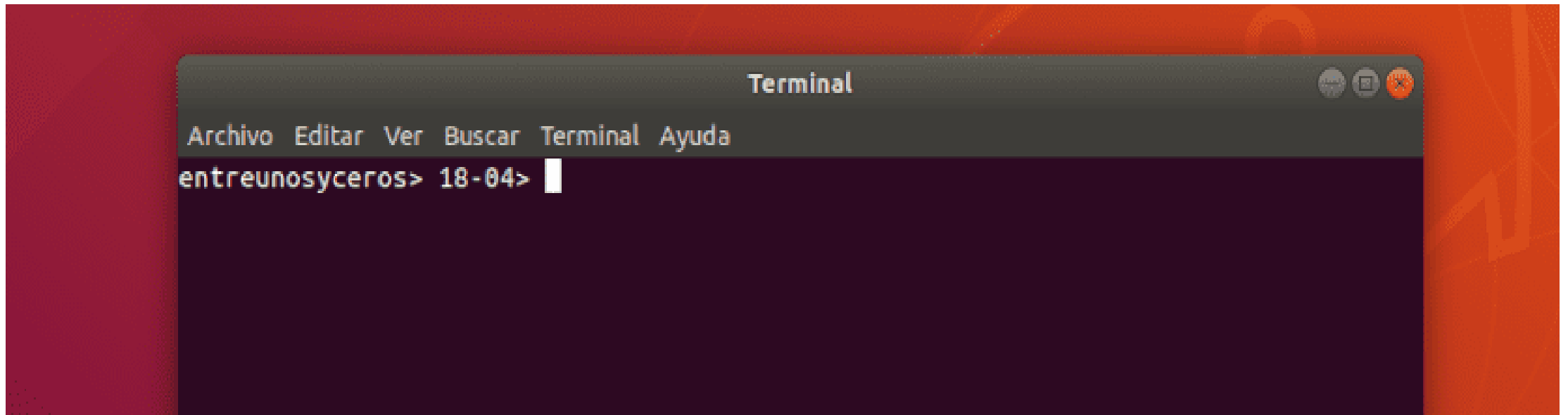
```
export PS1="\u "
```

Aquí, `\u` es la secuencia de escape.

Mostrar solo el nombre de usuario

Existen algunos valores más para agregar a la variable PS1 y poder cambiar esto. Hay que recordar que después de añadir algún cambio, debemos ejecutar el comando `'source ~/.bashrc'` para que se apliquen los cambios.

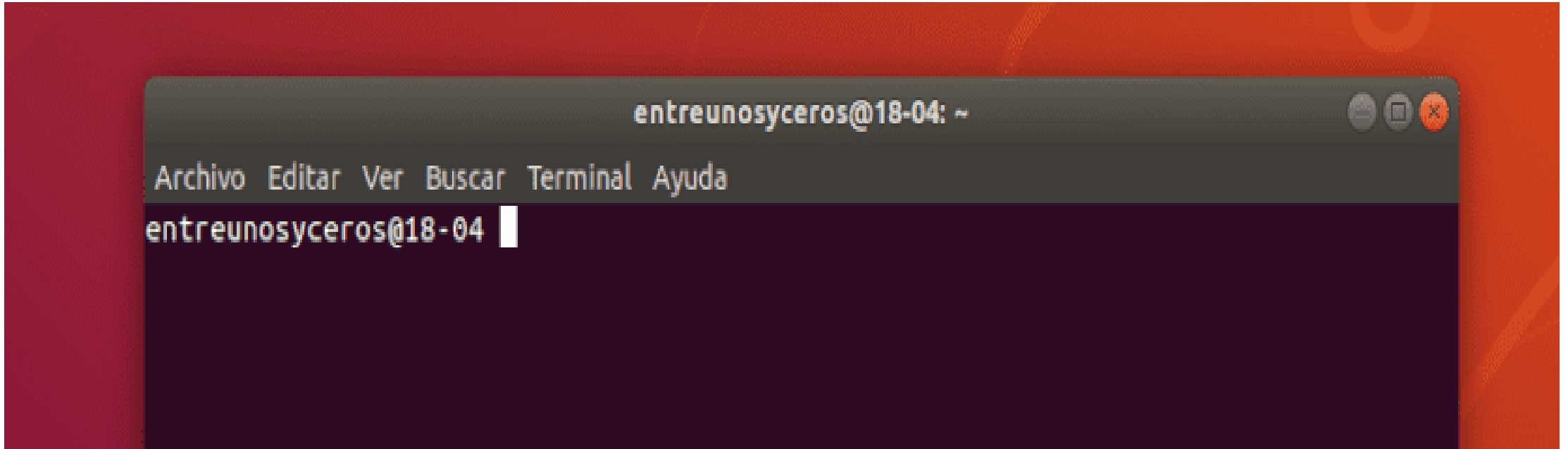
Añadir nombre de usuario con nombre de host



```
export PS1="\u> \h> "
```

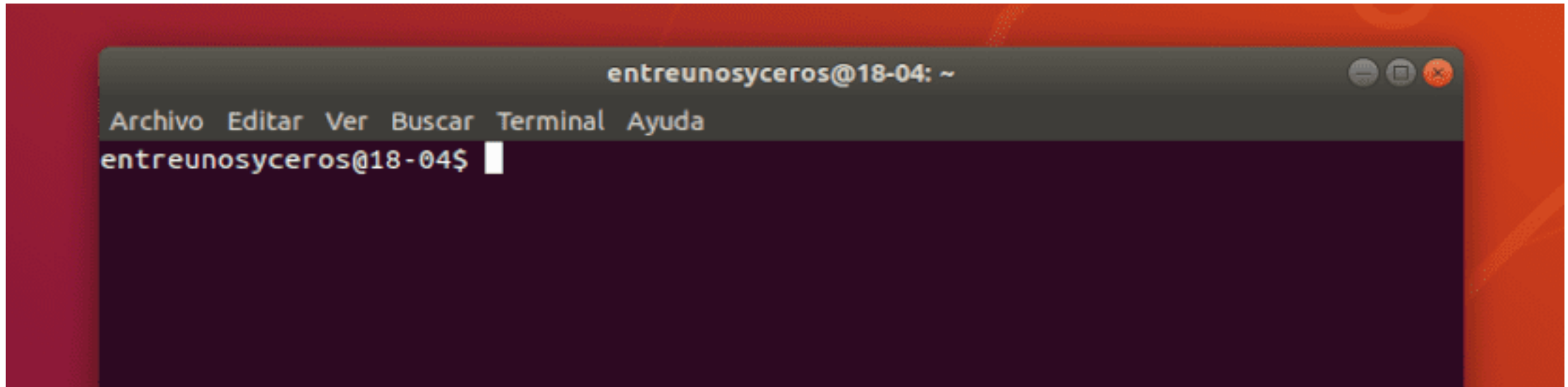

Agregar nombre de usuario y FQDN

Si quieres cualquier letra, por ejemplo @, entre el nombre de usuario y el nombre de host, utiliza la siguiente entrada:

A screenshot of a terminal window. The title bar shows 'entreunosyceros@18-04: ~'. Below the title bar is a menu bar with the following items: 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The main area of the terminal shows the prompt 'entreunosyceros@18-04' followed by a white cursor.

```
export PS1="\u@\h "
```

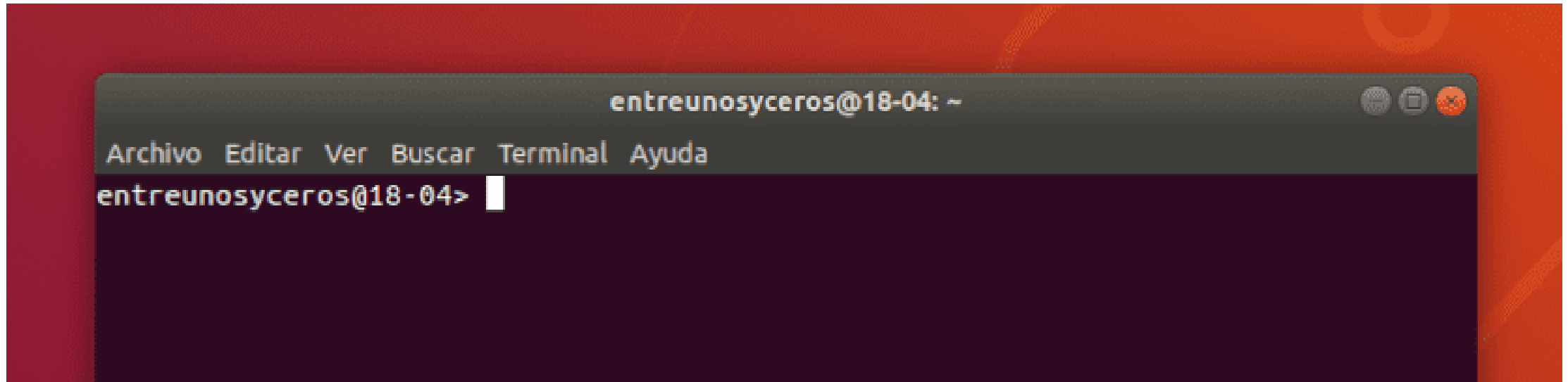
Añade el nombre de usuario con el nombre de host y el símbolo \$ al final

A screenshot of a terminal window. The title bar at the top reads "entreunosyceros@18-04: ~". Below the title bar is a menu bar with the options "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The main area of the terminal shows the prompt "entreunosyceros@18-04\$" followed by a white cursor block.

```
entreunosyceros@18-04: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
entreunosyceros@18-04$
```

```
export PS1="\u@\h\\$ "
```

Agregar caracteres especiales entre y después del nombre de usuario y el nombre de host

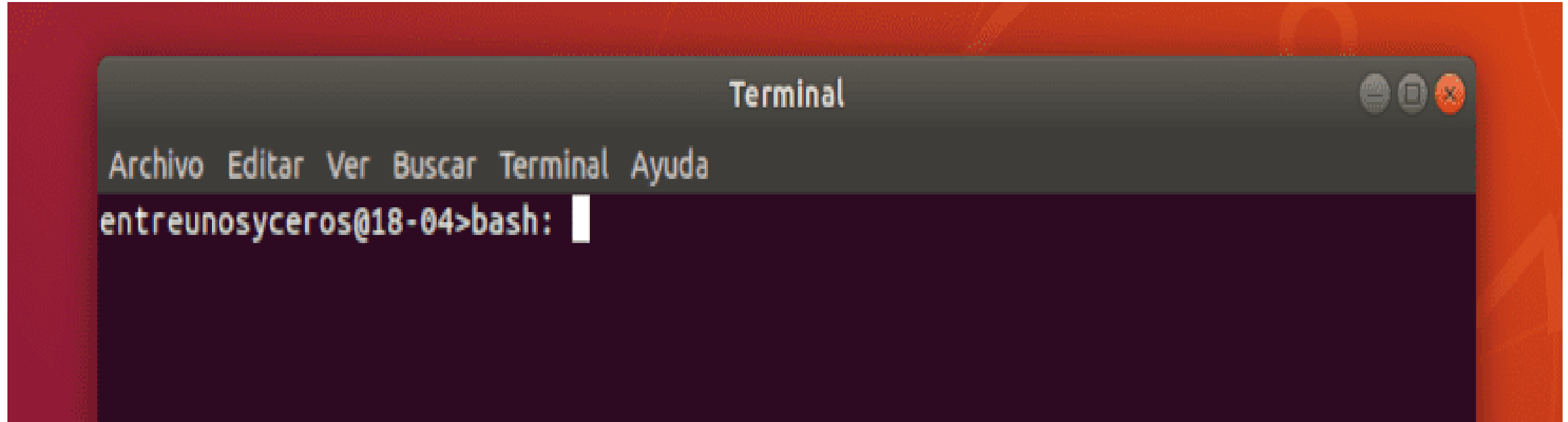
A screenshot of a terminal window with a dark background. The title bar at the top reads 'entreunosyceros@18-04: ~'. Below the title bar is a menu bar with the options 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The main area of the terminal shows the prompt 'entreunosyceros@18-04>' followed by a white cursor bar.

```
entreunosyceros@18-04: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
entreunosyceros@18-04> |
```

```
export PS1="\u@\h> "
```

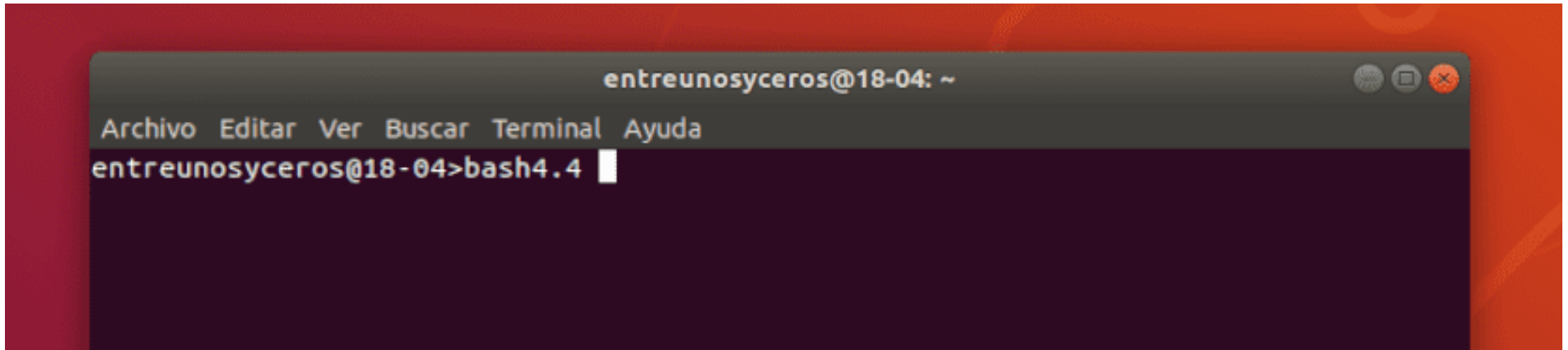
Del mismo modo, se pueden agregar otros caracteres especiales, como dos puntos, punto y coma, *, subrayado, espacio, etc.

Mostrar nombre de usuario, nombre de host, nombre de shell



```
export PS1="\u@\h>\s: "
```

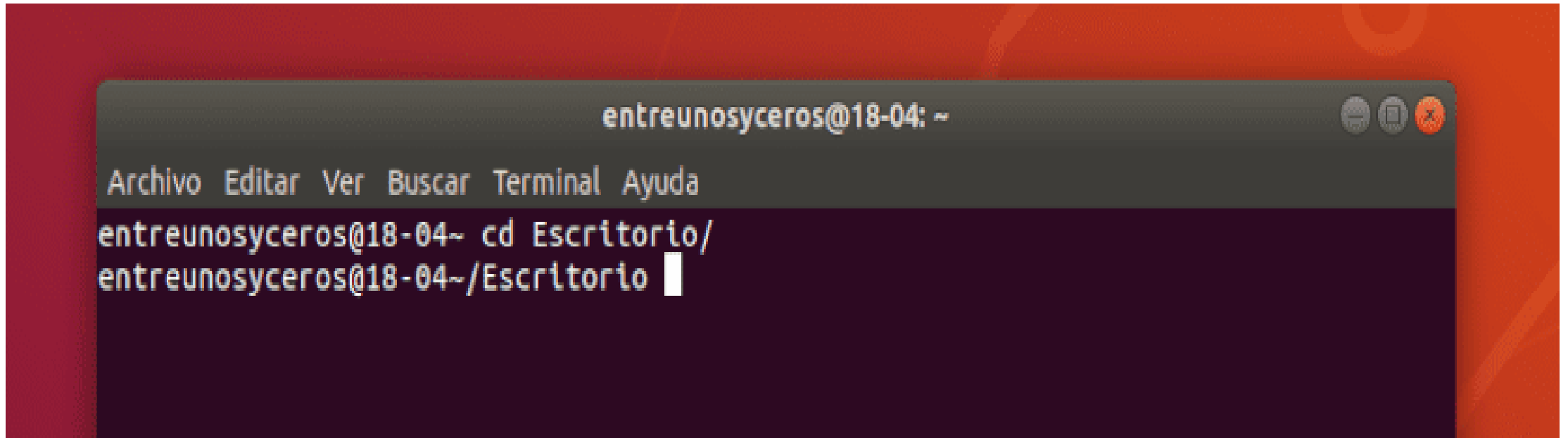
Mostrar nombre de usuario, nombre de host, shell y su versión

A screenshot of a terminal window titled 'entreunosyceros@18-04: ~'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The prompt 'entreunosyceros@18-04>' is followed by the command 'bash4.4' and a cursor. The terminal background is dark purple, and the window title bar is dark gray with standard Linux window controls on the right.

```
entreunosyceros@18-04: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
entreunosyceros@18-04>bash4.4
```

```
export PS1="\u@\h>\s\v "
```


Mostrar nombre de usuario, nombre de host y ruta al directorio actual

A screenshot of a terminal window with a dark background. The window title bar shows 'entreunosyceros@18-04: ~' and standard window control buttons. The menu bar includes 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal content shows the command 'cd Escritorio/' being entered, followed by the updated prompt 'entreunosyceros@18-04~/Escritorio' with a cursor at the end.

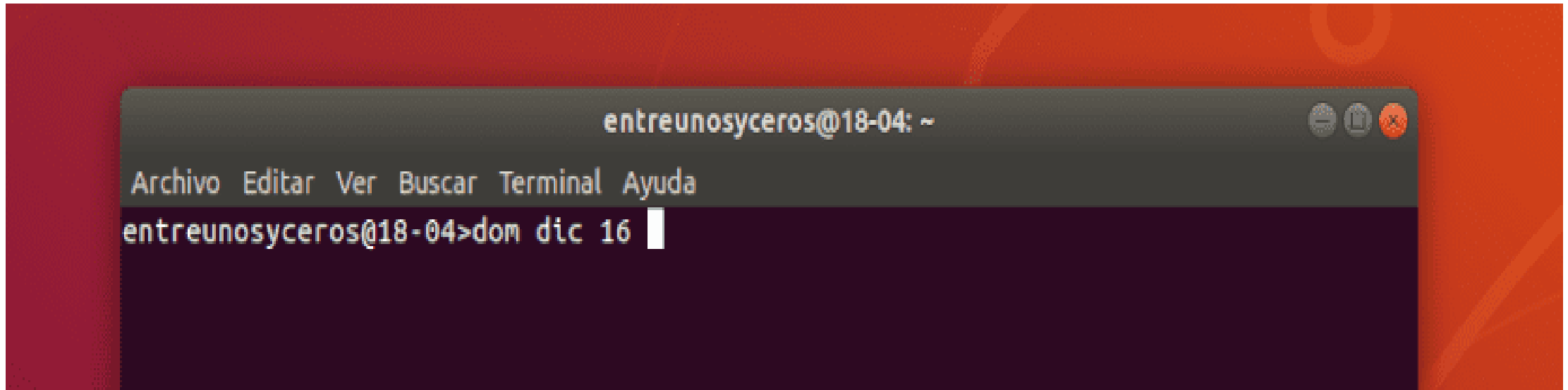
```
entreunosyceros@18-04: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
entreunosyceros@18-04~ cd Escritorio/  
entreunosyceros@18-04~/Escritorio
```

`export PS1="\u@\h\w "`

Verás el símbolo ~ si el directorio actual es \$HOME.

Visualizar la fecha en el mensaje de BASH

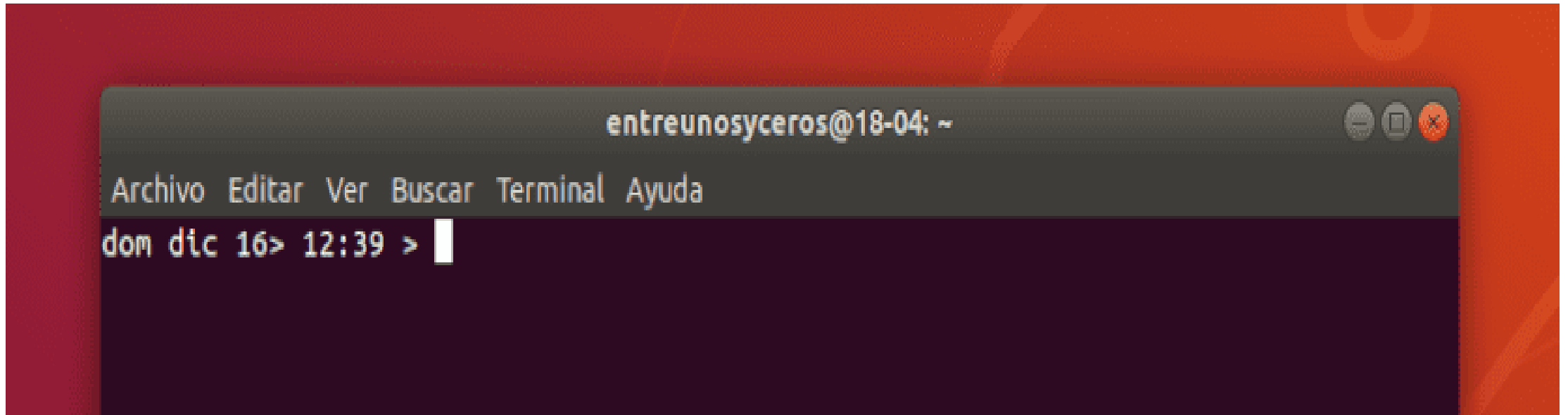
Para **mostrar la fecha con tu nombre de usuario y nombre del host** en el prompt, agrega lo siguiente en el archivo `~/.bashrc`.

A screenshot of a terminal window with a dark background. The title bar at the top reads 'entreunosyceros@18-04: ~'. Below the title bar is a menu bar with the options 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The main area of the terminal shows the prompt 'entreunosyceros@18-04>' followed by the command 'date' and a cursor. The output of the command is 'dom dic 16', which is partially visible on the line.

```
entreunosyceros@18-04: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
entreunosyceros@18-04>date  
dom dic 16
```

```
export PS1="\u@\h>\d "
```

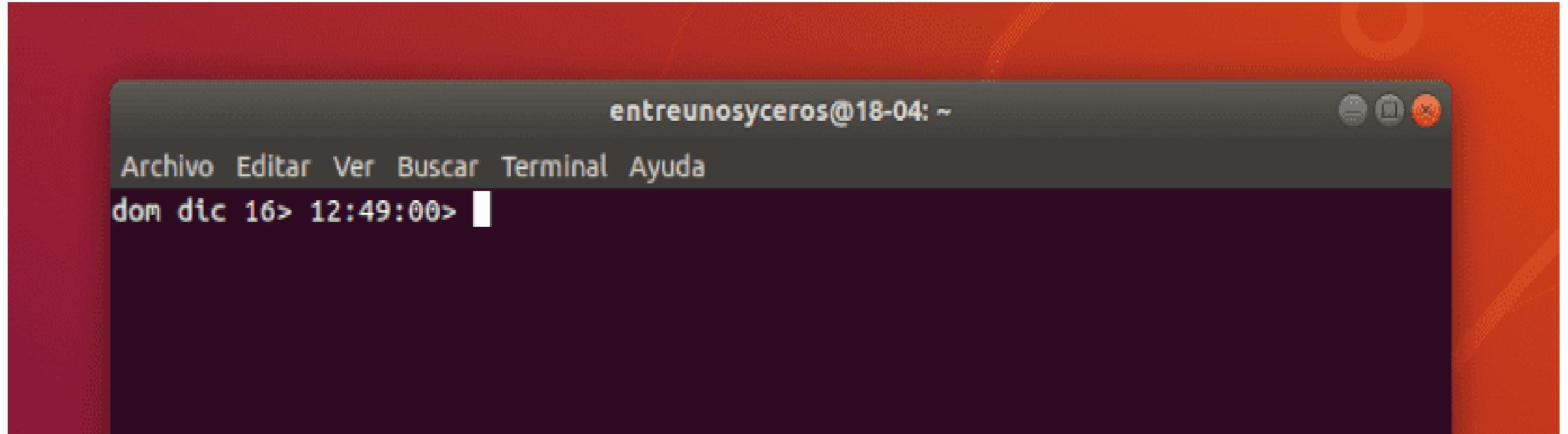
Fecha y hora en formato de 12 horas en BASH

A screenshot of a terminal window titled 'entreunosyceros@18-04: ~'. The window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal output shows 'dom dic 16> 12:39 >' followed by a cursor. The background of the terminal is dark purple, and the window title bar is dark gray with standard window controls on the right.

```
entreunosyceros@18-04: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
dom dic 16> 12:39 > |
```

```
export PS1="\d> \@ > "
```

Fecha y hora 12 formato hh:mm:ss



```
export PS1="\d> \T> "
```

Fecha y hora 24 horas

```
export PS1="\d> \A> "
```

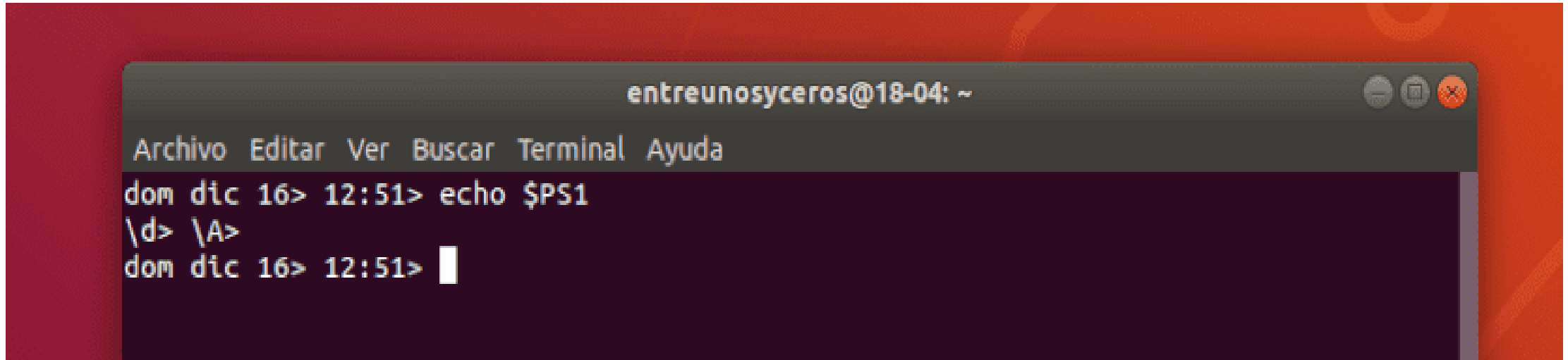
Fecha y hora en formato 24 horas hh:mm:ss

```
export PS1="\u@\h> \d\t "
```

Estas **son solo algunas secuencias de escape comunes** para cambiar el formato del mensaje Bash. Hay algunas secuencias más disponibles.

Fecha y hora en formato 24 horas hh:mm:ss

En cualquier momento se puede **ver la configuración actual** de la variable `$PS1` escribiendo en la terminal el comando:

A screenshot of a terminal window with a dark background. The title bar at the top reads 'entreunosyceros@18-04: ~'. Below the title bar is a menu bar with the options 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows a prompt 'dom dic 16> 12:51>' followed by the command 'echo \$PS1'. The output of the command is displayed on the next line as '\d> \A>'. Below the output, the prompt 'dom dic 16> 12:51>' is shown again with a white cursor character at the end.

```
entreunosyceros@18-04: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
dom dic 16> 12:51> echo $PS1  
\d> \A>  
dom dic 16> 12:51> █
```

`echo $PS1`

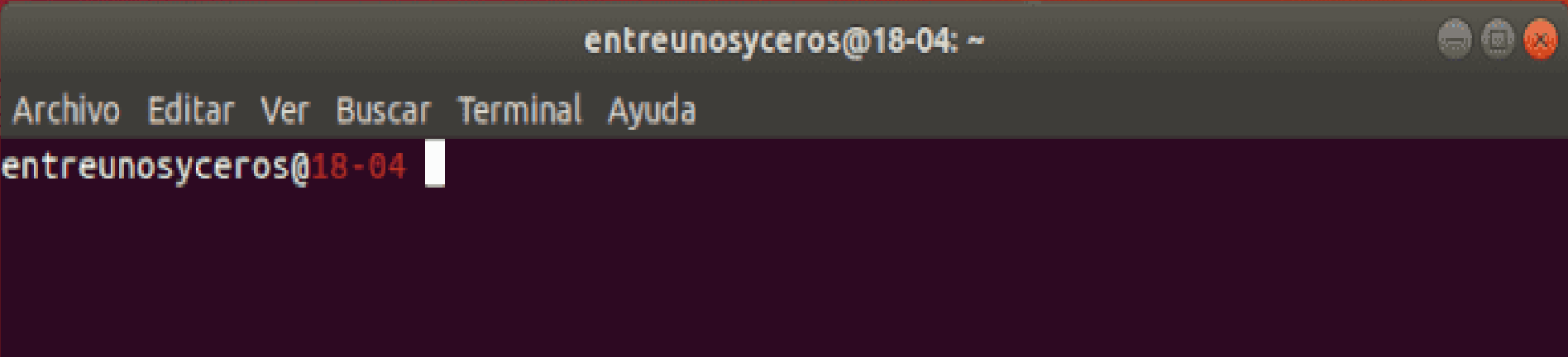
Coloreando el prompt

Lo que hemos visto hasta ahora es que acabamos de cambiar / agregar algunos elementos al bash prompt. Ahora vamos a modificar los colores a algunos elementos.

Para darle un toque de color al primer plano (texto) y color de fondo a los elementos, como hasta ahora **agregaremos un código al archivo ~/.bashrc.**

Coloreando el prompt

Por ejemplo, para cambiar el color del texto al nombre de host y que pase a verse rojo, agregaremos el siguiente código:

A screenshot of a terminal window with a dark background. The title bar at the top reads 'entreunosyceros@18-04: ~'. Below the title bar is a menu bar with the options 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The main area of the terminal shows the prompt 'entreunosyceros@18-04' in red text, followed by a white cursor. The terminal window is set against a red background with a faint circular pattern.

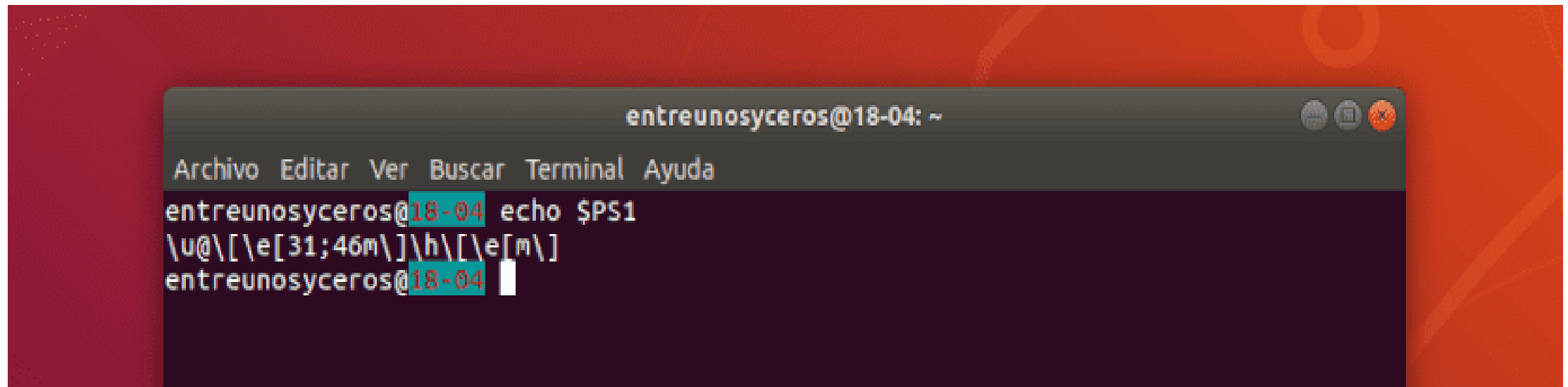
```
export PS1="\u@\[\e[31m\]\h\[\e[m\] "
```

Coloreando el prompt

Una vez agregado, hay que **actualizar los cambios** con el comando en la terminal:

```
source ~/.bashrc
```

Del mismo modo, para **cambiar el color de fondo al nombre del host**, el código a utilizar sería este:

A screenshot of a terminal window titled 'entreunosyceros@18-04: ~'. The terminal has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The prompt 'entreunosyceros@18-04' is highlighted in red. The user enters the command 'echo \$PS1', and the output is '\u@\[\e[31;46m\]\h\[\e[m\]', where the host name part is highlighted in red. The prompt then returns to 'entreunosyceros@18-04' with a white cursor.

```
entreunosyceros@18-04: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
entreunosyceros@18-04 echo $PS1  
\u@\[\e[31;46m\]\h\[\e[m\  
entreunosyceros@18-04
```

```
export PS1="\u@\[\e[31;46m\]\h\[\e[m\] "
```

echo

- echo hola mundo
- echo “hola mundo”
- echo ‘hola mundo’

Variables

env

```
LOGNAME=usuario
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
W_Archivos H=2
HOME=/home/usuario
USERNAME=usuario
IM_CONFIG_PHASE=1
LANG=es_PE.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=
42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=
;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2
1;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31
ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wlm=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg
1;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=0
5:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35
.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv
1;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:
4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xsp
0;36:
XDG_CURRENT_DESKTOP=ubuntu:GNOME
VTE_VERSION=6003
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/4b056056_688f_4afe_9ed1_72f967e56cc7
INVOCATION_ID=6deec8737bce40328081b98617ae34f2
MANAGERPID=1954
GJS_DEBUG_OUTPUT=stderr
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xterm-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=usuario
GNOME_TERMINAL_SERVICE=:1.108
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=8:36284
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
```

Variables

- Nombre=Maurice
- echo \$Nombre
Maurice
- echo 'mi nombres es \$Nombre'
- echo "mi nombres es \$Nombre"



2.2 Uso de la línea de comandos para obtener ayuda

Páginas de ayuda

¿No sabes cuáles son los pasos siguientes? A este respecto, Linux ofrece en el terminal **páginas de ayuda y de documentación** como las Unix Man Pages (Manual de Unix) y las páginas de información de GNU, las cuales contienen una exhaustiva descripción de todos los programas, comandos, archivos de configuración, formatos de archivo y funciones del kernel. Con *whatis* y *apropos* en la categoría de páginas de ayuda, cuentas con programas de líneas de comando que permiten examinar las páginas del manual del sistema operativo por palabras clave.

apropos

Con *apropos* se examinan los títulos y las descripciones de las páginas del manual de tu sistema operativo por palabras clave. El programa de líneas de comandos muestra en el terminal todas las correspondencias incluyendo una breve descripción.

Sigue el siguiente esquema:

apropos [OPCIONES] PALABRA CLAVE

Este comando soporta diferentes opciones. Con la opción *-e(--exact)* delimitas la búsqueda a correspondencias exactas, pero también puedes utilizar comodines (*-w '*TÉRMINO DE BÚSQUEDA'*) y expresiones habituales (*-r*).

info

El comando *info* abre las páginas de información de GNU sobre un tema determinado. Normalmente equivalen al manual que se puede invocar con *man*, aunque a diferencia de este, contienen enlaces que facilitan la navegación en el manual.

Para abrir la documentación de lectura de GNU utiliza esta sintaxis:

info [OPCIÓN] TEMA

Si no indicas una opción ni un tema accedes al menú principal del manual de GNU.

man

Con man se abren las páginas del manual (man pages) de tu distribución de Linux en la consola.

Para ello se sigue el siguiente esquema:

man [OPCIÓN] TEMA

Las man pages de Linux se distribuyen en 10 secciones temáticas:

- (1) Comandos de usuario
- (2) Llamadas al sistema
- (3) Biblioteca de funciones del lenguaje de programación C
- (4) Formatos de archivo
- (5) Archivos de configuración
- (6) Juegos
- (7) Miscelánea
- (8) Comandos para administrar el sistema
- (9) Funciones del kernel
- (n) Comandos nuevos

man

Si, por ejemplo, quieres abrir el manual para un comando específico, se combina man con el nombre del comando:

```
man clear
```

O se concreta la búsqueda con el número de la sección correspondiente:

```
man 1 clear
```

En ambos casos se abre la página del manual por el comando clear. Para cerrarlo y volver a la consola utiliza la tecla [Q].

Si quieres explorar las páginas del manual de Linux por palabras clave dispones del comando apropos.

Con *pinfo* cuentas con una versión del programa *info* que sigue el modelo del explorador de líneas de comandos Lynx y muestra las páginas de información con los enlaces destacados con colores.

Para utilizar *pinfo* se sigue el mismo esquema sintáctico que con *info*:

pinfo [OPCIONES] TEMA

whatis

El programa *whatis* permite buscar por palabras clave en el manual. Para examinar el manual del sistema operativo en busca de una equivalencia exacta se llama al programa con el término de búsqueda. Si se encuentra una correspondencia, *whatis* responde con una breve descripción en el terminal.

whatis [OPCIONES] PALABRA CLAVE

whatis (-w '*PALABRA CLAVE') también soporta comodines y expresiones regulares (-r).

/usr/share/doc

```
usuario@usuario-VirtualBox: /usr/share/doc

usuario@usuario-VirtualBox:~/Escritorio$ cd /usr/share/doc
usuario@usuario-VirtualBox:/usr/share/doc$ ls
accountsservice      ibus-data            libkeyutils1         libxxf86vm1
acl                  ibus-gtk             libklibc             libyajl2
acpid                ibus-gtk3            libkmod2             libyaml-0-2
acpi-support         ibus-table           libkpathsea6         libyelp0
adduser              tio-sensor-proxy     libkrb5-26-heimdal   libzstd1
adwaita-icon-theme   im-config            libkrb5-3            linux-base
aisleriot            info                 libkrb5support0      linux-firmware
alsa-base            init                 libksba8             linux-generic
alsa-topology-conf   initramfs-tools      liblangtag1          linux-headers-
alsa-ucm-conf         initramfs-tools-bin  liblangtag-common    linux-headers-
alsa-utils           initramfs-tools-core liblcms2-2            linux-headers-
amd64-microcode      init-system-helpers liblcms2-utils        linux-hwe-5.8-
anacron              inputattach          libldap-2.4-2         linux-hwe-5.8-
apg                 install-info         libldap-common        linux-image-5.
apparmor            intel-microcode      libldb2              linux-image-5.
app-install-data-partner ippusbxd             liblirc-client0       linux-image-g
apport              iproute2             libllvm11             linux-modules-
apport-gtk          iptables             libllvm9              linux-modules-
```

--help

```
usuario@usuario-VirtualBox:/usr/share/doc$ ls --help
```

Modo de empleo: ls [OPCIÓN]... [FICHERO]...

Muestra información acerca de los FICHEROs (del directorio actual por defecto).
Ordena las entradas alfabéticamente si no se especifica ninguna de las opciones -cftuvSUX ni --sort.

Los argumentos obligatorios para las opciones largas son también obligatorios para las opciones cortas.

-a, --all	no oculta las entradas que comienzan con .
-A, --almost-all	no muestra las entradas . y .. implícitas
--author	con -l, imprime el autor de cada fichero
-b, --escape	imprime escapes en estilo C para los caracteres no gráficos
--block-size=SIZE	with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below
-B, --ignore-backups	do not list implied entries ending with ~

man

man ls

```
usuario@usuario-VirtualBox: /usr/share/doc
LS(1) User Commands
NAME
    ls - list directory contents
SYNOPSIS
    ls [OPTION]... [FILE]...
DESCRIPTION
    List information about the FILES (the current directory by default). Sort entries alphabetically if
    not specified.

    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
        with -l, print the author of each file

    -b, --escape
        print C-style escapes for nongraphic characters

    --block-size=SIZE
        with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below

    -B, --ignore-backups
        do not list implied entries ending with ~

    -c
        with -lt: sort by, and show, ctime (time of last modification of file status information);
        otherwise: sort by ctime, newest first
```

Info ls

```
usuario@usuario-VirtualBox: /usr/share/doc

Next: dir invocation, Up: Directory listing

10.1 'ls': List directory contents
=====

The 'ls' program lists information about files (of any type, including
directories). Options and file arguments can be intermixed arbitrarily,
as usual.

For non-option command-line arguments that are directories, by
default 'ls' lists the contents of directories, not recursively, and
omitting files with names beginning with '.'. For other non-option
arguments, by default 'ls' lists just the file name. If no non-option
argument is specified, 'ls' operates on the current directory, acting as
if it had been invoked with a single argument of '.'.

By default, the output is sorted alphabetically, according to the
locale settings in effect.(1) If standard output is a terminal, the
output is in columns (sorted vertically) and control characters are
output as question marks; otherwise, the output is listed one per line
and control characters are output as-is.

Because 'ls' is such a fundamental program, it has accumulated many
options over the years. They are described in the subsections below;
within each section, options are listed alphabetically (ignoring case).
The division of options into the subsections is not absolute, since some
options affect more than one aspect of 'ls''s operation.

Exit status:

0 success
1 minor problems (e.g., failure to access a file or directory not
specified as a command line argument. This happens when listing a
directory in which entries are actively being removed or renamed.)
2 serious trouble (e.g., memory exhausted, invalid option, failure
```




2.3 Uso de directorios y archivos de listas

operaciones en el directorio

Algunos comandos básicos de Linux permiten llevar a cabo operaciones en los directorios del sistema, como puede ser crear ficheros, borrarlos y gestionarlos, así como navegar por el árbol del directorio. Entre los comandos más importantes de esta categoría se incluyen `cd`, `ls`, `mkdir` o `rmdir`.

cd

El comando *cd* es la abreviatura de *change directory* y se utiliza para navegar por el directorio.

La sintaxis de esta orden sigue el esquema:

cd [OPCIÓN] DIRECTORIO

Si no se indica ningún directorio concreto, *cd* cambia automáticamente al directorio principal del usuario.

Si se acompaña de un guion (-), *cd* abre el directorio precedente.

chroot

La orden *chroot* (abreviatura de *change root*) se utiliza para ejecutar un comando en un directorio raíz diferente para, por ejemplo, aislar programas peligrosos del sistema de archivos restante, lo que se denomina *chroot jail*.

Este comando requiere derechos raíz y se orienta por este esquema:

chroot DIRECTORIO COMANDO

ls

La orden *ls* equivale a *list* y se utiliza para mostrar el contenido de un fichero (los nombres de todos sus archivos y carpetas).

Esta es la sintaxis de la orden:

ls [OPCIONES] DIRECTORIO

Si no se añade a *ls* ningún directorio, el comando enumera el contenido del directorio en curso.

Con ayuda de diferentes opciones se puede definir qué información se ha de mostrar y cómo.

mkdir

El comando *mkdir* corresponde a *make directory* y permite a los usuarios de Linux crear directorios desde cero.

Para crear un directorio en el fichero en curso escribe la siguiente sintaxis:

mkdir [OPCIÓN] DIRECTORIO

Si lo que se necesita es crear varios ficheros a la vez, se escribe uno detrás de otro sin signos de puntuación y con espacio intermedio:

mkdir [OPCIÓN] DIRECTORIO1 DIRECTORIO2

Si se quiere crear un directorio nuevo en otro fichero diferente al actual se ha de indicar la ruta absoluta o relativa al fichero:

mkdir /home/user/Desktop/test

mkdir/Desktop/test

En ambos ejemplos se crea el directorio *test* en el directorio *Desktop*.

mkdirhier

Con *mkdirhier* puedes crear jerarquías completas con un único comando:

mkdirhier [OPCIÓN] /home/user/directorio1/directorio2/directorio3

Si *directorio1* y *directorio2* ya existieran, entonces *mkdirhier* solo crearía *directorio3*. En caso contrario, se crean los tres.

pwd

Con *pwd* (abreviatura de *print working directory*) la consola muestra el nombre del directorio de trabajo (en curso).

La sintaxis del comando es:

pwd [OPCIONES]

rmdir

Si hiciera falta borrar un determinado directorio, se utiliza el comando *rmdir* (*remove directory*) según la siguiente sintaxis:

rmdir [OPCIÓN] DIRECTORIO

Con *rmdir*, sin embargo, solo se pueden borrar directorios vacíos. Si se quiere eliminar un fichero con todas sus carpetas y subcarpetas se utiliza el comando *rm* (*remove*) con la opción *-r*.

Cuidado: *rmdir* no solicita confirmar el borrado. Los directorios seleccionados con el comando se borran definitivamente.

tree

Mientras que *ls* muestra el contenido de los directorios como lista, el comando *tree*, siguiendo este esquema sintáctico, muestra la jerarquía completa del directorio en forma de árbol:

tree [OPCIONES] [DIRECTORIO]



2.4 Crear, mover y eliminar archivos

Operaciones en archivos

Los comandos Linux de esta categoría permiten llevar a cabo diversas operaciones desde el terminal que atañen a los archivos y, utilizando comandos básicos como *cp*, *mv* y *rm*, se pueden copiar, desplazar, renombrar o borrar archivos del sistema.

basename

Al comando *basename* se le indica una ruta a un archivo y devuelve su nombre sin la ruta.

La sintaxis del comando se compone de:

basename [OPCIONES] Ruta/al/archivo [SUFIJO]

Si en el terminal escribes, por ejemplo, *\$ basename /home/user/imagen.jpg*, obtienes la siguiente respuesta:

imagen.jpg

Si se añade el sufijo de nuevo, el terminal no lo devuelve, como vemos aquí:

Entrada: *\$ basename /home/user/imagen.jpg .jpg*

Salida: *imagen*

El comando puede ser ampliado a varios archivos con diferentes opciones.

cat

El comando *cat* (de *concatenate*) nace como herramienta para enlazar archivos y puede emplearse como pager para mostrar el contenido de los archivos en el terminal.

Escribe *cat* con la siguiente sintaxis para leer y mostrar un archivo en la salida estándar:

```
cat OPCIONES ARCHIVOS
```

Si incluyes varios archivos, sepáralos mediante espacios:

```
cat OPCIONES ARCHIVO1 ARCHIVO2
```

Para concatenar el contenido de varios archivos se utilizan los operadores de redirección `>`, `<` y `|`. Si utilizas el operador “mayor que” (`>`), se aúna el contenido de dos archivos en un tercero:

```
cat archivo_1.txt archivo_2.txt > archivo_3.txt
```

cmp forma parte del paquete *diff* y, como este, también se utiliza para comparar el contenido de varios archivos aunque, a diferencia de como sucede en *diff*, aquí el cotejo tiene lugar byte por byte, lo que lo hace especialmente idóneo para archivos binarios.

Invoca a *cmp* según la siguiente sintaxis:

```
cmp [OPCIONES] ARCHIVO1 ARCHIVO2
```

Si *cmp* encuentra diferencias, entrega el número de bytes y el de la línea del primer archivo que difiera.

comm

Utiliza el comando *comm* para comparar por líneas archivos previamente ordenados (con *sort*, p. ej.).

La invocación de este comando se fundamenta en la siguiente estructura sintáctica:

comm [OPCIONES] ARCHIVO1 ARCHIVO2

Sin opciones adicionales, el programa genera una salida en tres columnas: la primera contiene todas las líneas que solo aparecen en *ARCHIVO1*; la segunda, todas las líneas que solo aparecen en *ARCHIVO2*, y la tercera, todas las líneas que aparecen en ambos archivos.

El comando *comm* soporta tres opciones:

- 1 = ocultar líneas únicas del *ARCHIVO1*
- 2 = ocultar líneas únicas del *ARCHIVO2*
- 3 = ocultar todas las líneas que aparecen en ambos archivos

La orden *cp* (de *copy*) se utiliza para copiar archivos o ficheros y sigue la siguiente sintaxis:

cp [OPCIONES] ORIGEN DESTINO

El elemento ORIGEN es el que se ha de copiar y DESTINO se define a un archivo o un directorio donde se ha de alojar el elemento copiado. Si se define como destino a un archivo que ya existe, el archivo origen reescribe su contenido. También se puede crear un archivo de destino nuevo. Si se han de copiar varios archivos, entonces el destino ha de ser un directorio, del mismo modo que si se copia un directorio.

Copiar un archivo de origen en un archivo de destino en el directorio actual:

cp [OPCIONES] ORIGEN DESTINO

Ejemplo: *cp archivo.txt archivo_copia.txt*

Copiar un archivo del directorio actual en un directorio de destino:

cp [OPCIONES] ARCHIVO_ORIGEN DIRECTORIO_DESTINO

Ejemplo: *cp archivo.txt home/user/documentos/2017*

Copiar varios archivos en un directorio:

cp [OPCIONES] ARCHIVO_ORIGEN1 ARCHIVO_ORIGEN2 DIRECTORIO_DESTINO

Ejemplo: *cp archivo.txt archivo.odt home/user/documentos/2017*

Copiar un directorio desde el directorio actual en otro directorio diferente:

cp DIRECTORIO_ORIGEN DIRECTORIO_DESTINO

Ejemplo: *cp directorio1 home/user/documentos/2017*

Si se tienen que copiar todos los directorios con el contenido completo se deben incluir en el copiado todos los subdirectorios con la opción -R.

cut

La orden *cut* permite extraer el contenido de las líneas de texto de un archivo (log o CSV) por columnas.

La sintaxis de este comando es:

cut [OPCIONES] ARCHIVO

Para indicar la posición exacta del fragmento que se tiene que extraer se utilizan las opciones *-b* (posición de byte), *-c* (posición del carácter), *-d* (carácter de separación) y *-f* (campo).

diff

Con el programa *diff* se cotejan dos archivos. También permite averiguar si dos directorios contienen los mismos archivos. Se le invoca con esta estructura:

diff [OPCIONES] ARCHIVO1 ARCHIVO2

dirname

dirname es lo contrario que *basename*: el comando permite extraer la parte de la ruta a un fichero sin mostrar su nombre.

Su sintaxis es:

dirname [OPCIONES]

Introduce `$ dirname /home/user/imagen.jpg` en el terminal para obtener la siguiente salida:

/home/user

file

Con *file* se puede obtener información sobre el tipo o formato de un archivo.

Sigue el siguiente esquema:

file [OPCIONES] ARCHIVO

In

El programa *ln* (*link*) crea un vínculo (un archivo especial) que lleva a un archivo o un directorio. Con ello se genera otra entrada a este archivo que permite acceder a él a través de una ruta diferente. Para invocar *ln* el comando ha de incluir, como mínimo, la ruta al archivo de origen.

ln [OPCIONES] ruta/al/archivo de origen

En este caso, el enlace se crea en el directorio actual de trabajo sin cambiar de nombre. Otra alternativa consiste en indicar una ruta de destino y dar un nombre diferente al vínculo:

ln [OPCIONES] ruta/al/archivo de origen ruta/al/enlace

In

ln crea por defecto los llamados enlaces duros o hardlinks, que no son adecuados para crear vínculos a los directorios y tampoco se pueden utilizar más allá de las fronteras de partición. Es por esto que el comando suele usarse en combinación con la opción *-s* (*--symbolic*), con la cual se pueden generar enlaces simbólicos, incluso más allá de los límites del sistema de archivos. Los enlaces simbólicos se refieren siempre a la ruta “real” a un archivo y dependen de esta.

Isof

Isof equivale a *list open files* (muestra los archivos abiertos), un programa de apoyo que muestra en el terminal información sobre los archivos abiertos ordenados en función del PID (ID de proceso).

Para invocarlo se sigue el siguiente esquema:

Isof [OPCIONES]

Dado que todos los sistemas tipo UNIX como Linux lo gestionan todo en archivos («*Everything is a file*»), la lista que devuelve *Isof* es muy larga. Por eso suelen usarse opciones que concretan el resultado.

md5sum

Con ayuda del comando *md5sum* se pueden calcular y comprobar sumas de control MD5.

El programa *mv* (*move*) copia un archivo o un directorio y borra el original. Si esto tiene lugar dentro del mismo directorio, *mv* puede utilizarse también para cambiar el nombre de los archivos.

Para ello se sigue el siguiente esquema:

mv [OPCIÓN] ORIGEN DESTINO

Ejemplos de su aplicación:

Trasladar un archivo a otro directorio:

mv [OPCIONES] ARCHIVO_ORIGEN DIRECTORIO_DESTINO

Ejemplo: *mv archivo1.txt home/user/documentos/2017*

Trasladar varios archivos de origen a un directorio de destino:

mv [OPCIONES] ARCHIVO_ORIGEN1 ARCHIVO_ORIGEN2 DIRECTORIO_DESTINO

Ejemplo: *mv archivo1.txt archivo2.txt home/user/documentos/2017*

Trasladar un subdirectorio del fichero actual a un directorio de destino:

mv [OPCIONES] DIRECTORIO_ANTIGUO DIRECTORIO_NUEVO

Ejemplo: *mv directorio1 home/user/documentos/2017*

Cambiar el nombre de un archivo en el directorio de trabajo:

mv [OPCIONES] ARCHIVO_ANTIGUO ARCHIVO_NUEVO

Ejemplo: *mv archivo1.txt archivo2.txt*

Cambiar el nombre de un subdirectorio en el directorio actual:

mv [OPCIONES] DIRECTORIO_ANTIGUO DIRECTORIO_NUEVO

Ejemplo: *mv directorio1 directorio2*

paste

De forma parecida a *cat*, el programa *paste* también muestra el contenido de los archivos en la salida estándar, pero mientras que *cat* los muestra de forma encadenada, *paste* los agrupa en columnas.

El esquema básico de esta función es:

```
paste [OPCIONES] ARCHIVO1 ARCHIVO2
```

En el modo estándar, los archivos indicados se agrupan de tal forma que todas las líneas cuyo número coincide se emiten en una misma línea en la salida. Así, cada línea de la salida contiene el contenido de todos los archivos de entrada.

Con la opción *-d* se puede configurar individualmente el tipo de separador que se usa. Por defecto suele utilizarse la tabulación.

Con la opción *-s* (de serial) se activa otro modo diferente en el cual todas las líneas del primer archivo de entrada se pegan en la primera línea de la salida. Los datos del resto de archivos de entrada se pegan a continuación en líneas de salida separadas. De esta forma, cada línea de la salida contiene únicamente el contenido de un archivo de entrada.

rename

rename permite cambiar el nombre de archivos y carpetas con ayuda de expresiones regulares compatibles con perl (Regex). Al contrario de *mv*, *rename* se presta así para modificar los nombres de varios archivos parcial o totalmente.

La invocación de *rename* se realiza siguiendo el siguiente esquema:

rename [OPCIONES] 'EXPRESIÓN_REGULAR' ARCHIVOS

En el caso de las sustituciones, las expresiones regulares siguen la siguiente sintaxis:

s/PATRONDEBUSQUEDA/SUSTITUCIÓN/MODIFICADOR

En el siguiente ejemplo todas las terminaciones *.html* se sustituyen por *.xhtml*:

*rename 's/\.html\$/\.xhtml/' *.html*

El programa *rm* (*remove*) borra archivos o directorios de forma irreversible. Para ello se sigue la siguiente estructura:

rm [OPCIONES] ARCHIVO

o

rm [OPCIONES] DIRECTORIO

Para eliminar un directorio junto a todos sus subdirectorios se utiliza *rm* con la opción *-R* (*--recursive*):

rm -R DIRECTORIO

Cuando se ordena eliminar varios archivos o ficheros, estos se separan por espacios:

rm [OPCIONES] ARCHIVO1 ARCHIVO2

shred

shred es un comando que permite borrar archivos de forma permanente porque los reescribe de modo que, incluso contando con los mejores medios forenses, no se pueden restablecer.

Este comando se invoca siguiendo este esquema básico:

shred [OPCIONES] ARCHIVO

Utilizándolo con las siguientes opciones se eliminan archivos concretos:

shred -fuz ARCHIVO

La opción *-f* impulsa el proceso de eliminación, *-z* reescribe el contenido del archivo con ceros (la configuración estándar son datos aleatorios) y *-u* elimina el archivo triturado del sistema de forma similar al comando *rm*.

sort

Con el comando *sort* se ordenan las listas de archivos y las salidas de programa en líneas numérica y alfabéticamente.

Se invoca a *sort* siguiendo este esquema:

sort [OPCIONES] ARCHIVO

Este método se puede complementar con opciones como *-n* (ordenar numéricamente), *-R* (ordenar de forma aleatoria) o *-r* (ordenar en orden inverso).

split

El comando *split* se utiliza para subdividir archivos. Sigue el siguiente esquema sintáctico:

split [OPCIONES] [INPUT [PREFIJO]]

INPUT equivale al archivo que se ha de subdividir y PREFIJO hace las veces de prefijo del nombre del nuevo subarchivo. A la hora de cambiar su nombre se sigue el siguiente esquema:

PREFIJOaa, PREFIJOab, PREFIJOac ...

Si no se define ningún prefijo, *split* recurre al estándar x.

Con la opción *-b* (bytes) se puede definir el tamaño de las partes en que se divide el archivo. Esto puede hacerse en bytes (b), kilobytes (k) o megabytes (m).

split

Ejemplo:

```
split -b 95m archivo.tgz split-archivo.tgz.
```

Aquí *split* divide el archivo *archivo.tgz* en archivos de 95 megabytes y los nombra como sigue:

```
split-archivo.tar.aa
```

```
split-archivo.tar.ab
```

```
split-archivo.tar.ac
```

Los archivos que se han subdividido utilizando *split* pueden reunirse de nuevo con el comando *cat*.

```
cat split-archivo.tar.* > archivo.tar
```

stat

Con ayuda del comando *stat* (*status*) se puede obtener la fecha y la hora de los accesos y los cambios relativos a determinados archivos o directorios.

Este comando sigue esta sintaxis:

stat [OPCIONES] ARCHIVO

El formato de salida se puede configurar con opciones.

touch

Para modificar la fecha y la hora de los accesos o los cambios que han tenido lugar en un archivo se utiliza el comando *touch*. Si se utiliza en un archivo que no existe, este se crea automáticamente, lo que hace que este comando también se utilice para generar archivos vacíos.

Para invocar a *touch* se tiene que seguir este esquema:

touch [OPCIONES] ARCHIVO

Si de lo que se trata es de cambiar la fecha y la hora de un archivo por una fecha determinada se utiliza la opción *-t* incluyendo la fecha en el formato *[AA]MMDDhhmm[.ss]*.

Ejemplo:

touch -t 1703231037 archivo.txt

La fecha y la hora del acceso y los cambios respecto a este archivo se han fijado en el 23 de marzo de 2017 a las 10:37 horas. Si se prefiere, el cambio se puede limitar a la fecha de acceso o de modificación con las opciones *-a* y *-m*. Si se utiliza sin la opción *-t* se sella automáticamente con la fecha y la hora actual.

El comando *uniq* suele usarse junto con *sort* para limpiar líneas dobles en los archivos ordenados.

En el siguiente ejemplo se enlaza el comando *sort* con el comando *uniq* con la barra vertical para seleccionar en primer lugar un archivo y a continuación para emitirlo sin líneas duplicadas:

```
sort archivo.txt | uniq
```

CENTRO NACIONAL DE SEGURIDAD DIGITAL

