

Introducción a Linux

Módulo 2



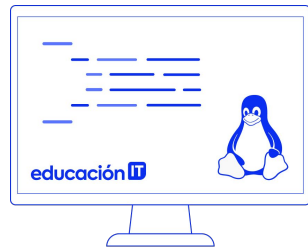
Comenzando a usar el shell

Empezando a usar el shell

Una vez que el usuario se ha identificado dentro del sistema Linux, se enfrenta con el **shell**, que aparece simplemente como **una interfaz de línea de comandos**.

El shell cuenta con un indicador para ingresar comandos llamado prompt, esos comandos son interpretados por el shell y enviados al sistema. El shell que se esté corriendo en ese momento configura su prompt correspondiente.

La mayoría de los sistemas Linux tienen como predeterminado el shell **bash** que, generalmente, está configurado para mostrar el nombre de usuario, nombre del servidor y directorio actual de trabajo en el prompt.



Un ejemplo de este prompt sería:

```
[educacionit@centos /etc]$
```

En este ejemplo, **educacionit** es el nombre de usuario, **centos** es el nombre del servidor y **/etc** es el directorio actual de trabajo (llamado **pwd** – present working directory). El prompt para el shell de bash es el símbolo **\$** para un usuario común y **#** para el usuario root. Al prompt lo configuramos a través del archivo **/etc/bashrc** o el archivo **/etc/bash.bashrc** (este último presente en distribuciones basadas en Debian).

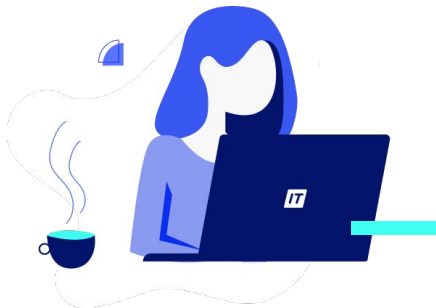
Por medio de este archivo, podemos cambiar la configuración de lo que se muestra en el prompt.

Cada usuario puede personalizar el prompt a su gusto, editando el archivo **~/.bashrc** o el archivo **~/.profile** (este último es más común en las distribuciones basadas en Debian).



Tipos de shell

Existen varios tipos de shell. Difieren ligeramente en sintaxis, funcionalidades y configuración. La mayoría de las distribuciones de Linux usan de manera predeterminada el shell bash.



Chequear el siguiente archivo para saber cuáles se tienen instaladas:

```
$ cat /etc/shells  
/bin/sh  
/bin/bash  
/sbin/nologin  
/bin/dash  
/bin/tcsh  
/bin/csh  
/usr/bin/irssi
```

Comandos y sintaxis

Comandos y sintaxis

Los comandos que se escriben en la línea de comandos no son enviados al sistema hasta que no se haya pulsado la tecla Enter. Esto permite editarlos antes de que se pasen al sistema.

Deben seguir una sintaxis específica: la primera palabra es el comando que se debe ejecutar. Puede incluir una ruta absoluta, que es la ruta completa al comando (debe terminar con el nombre del comando). A continuación, deben figurar las opciones que usa el comando llamadas parámetros. Los argumentos del comando van a continuación de las opciones.

Cada uno de estos elementos van separados, en la línea de comandos, por un espacio en blanco. El formato de un comando es el siguiente:

```
$ comando opciones argumentos
```

Recordatorio de examen: es importante recordar esta sintaxis. Algunas preguntas del examen son sobre el correcto uso de comandos, opciones y argumentos.

Por ejemplo, tomemos el comando **ls**, que lista los archivos y directorios que hay en el **pwd**. Para ejecutarlo, simplemente hay que teclear:

```
$ ls
```

Las opciones son códigos de una letra precedidos por un guión “-”, y modifican la acción del comando. Podemos combinar y usar varias opciones con un mismo comando. Las opciones son sensibles a mayúsculas y, en muchas ocasiones, una letra minúscula significa una opción diferente que su correspondiente mayúscula.

Una opción muy importante disponible con muchos comandos es la “-R” que especifica que el comando se debe ejecutar de forma recursiva a través del árbol de directorios (recursivo significa listar el contenido de los subdirectorios contenidos en el directorio del que se desea ver el contenido).

Recordatorio de examen: la opción para la función recursiva de un comando es muy importante y aparecerá en el examen.

Si se desea listar los archivos y los directorios, y que los directorios aparezcan seguidos de una barra inclinada (/), se puede usar la opción “-F”.

Prueba del resultado:

```
$ ls -F
```

Se desea mostrar el contenido del directorio **/etc** desde otra localización. Además, que se muestre una barra inclinada tras todos los nombres de directorios que estén dentro de **/etc**. El directorio **/etc** se puede utilizar como argumento para el comando **ls**.

Un **argumento** es otra opción que se puede utilizar con un comando. En el siguiente ejemplo, el argumento es un directorio que será examinado por el comando. Podríamos probar ejecutar el comando de la siguiente manera:

```
$ ls -F /etc
```

Otro carácter especial que podemos utilizar cuando introducimos comandos es la barra invertida (\). Cuando se introduce a la derecha, antes de pulsar la tecla Enter, la barra invertida permite extender el comando a lo largo de varias líneas. Veamos la próxima slide.

La barra invertida provoca que el sistema ignore la pulsación de la tecla Enter, y trata a todos los comandos como si estuvieran en una única línea. Esta función es muy útil si tecleamos comandos muy largos, para poder partirlos en varias líneas. Un ejemplo sería el siguiente:

```
$ ls -F /etc \ ls -F /usr
```

En el mundo real: para ejecutar comandos muy largos se utilizan scripts, en lugar de teclearlos directamente sobre la línea de comandos. Sin embargo, **es importante recordar el uso de la barra invertida para el examen.**

Veamos otras opciones del comando **ls**.

Devuelve la lista de archivos y directorios en formato extenso, incluyendo nombre, privilegios de acceso, propietario, tamaño, fecha de última modificación, etc.

```
$ ls -l
```

Muestra todos los archivos y directorios del pwd, incluyendo los archivos ocultos.

```
$ ls -a
```

También pueden combinarse:

```
$ ls -la
```

Concatenación de comandos

En todos los ejemplos vistos hasta ahora, luego de escribir un comando, se pulsó la tecla Enter para informarle al sistema que debía procesarlo. Sin embargo, se puede escribir más de un comando por vez.

Se pueden ejecutar varios comandos, sin que estén conectados entre sí, tipeándolos en la misma línea separados por punto y coma “;”. Por ejemplo, es posible listar todos los archivos del directorio actual y la fecha de hoy tecleando:

```
$ ls ; date
```

El **punto y coma** es un carácter especial que siempre significa que hay varios comandos en la misma línea. Debido a que esto último tiene un sentido global, podemos prescindir de los espacios en blanco a ambos lados del punto y coma para obtener el mismo resultado (**ls;date**). Es importante destacar que es un comando que termina independientemente de cómo finaliza y se ejecuta a continuación el otro.

También podemos ejecutar varios comandos a la vez con los operadores lógicos `&&` , `||` . En este caso, solo ejecuta el comando si cumple con la función lógica dado que, cuando un comando se ejecuta, devuelve un estado `0` -bien- u otro valor mayor que `0` que se interpreta como error.

```
$ ls /pepe && date  
ls: cannot access /pepe: No such file or directory
```

Ahí no se muestra la fecha porque falló el primer comando.

```
$ ls -ld /etc && date  
drwxr-xr-x. 171 root root 12288 Oct 27 14:49 /etc  
Thu Oct 27 16:06:38 ART 2011
```

Se puede ver que al funcionar el primer comando, posteriormente se ejecuta el otro.



**¡Sigamos
trabajando!**