HFC

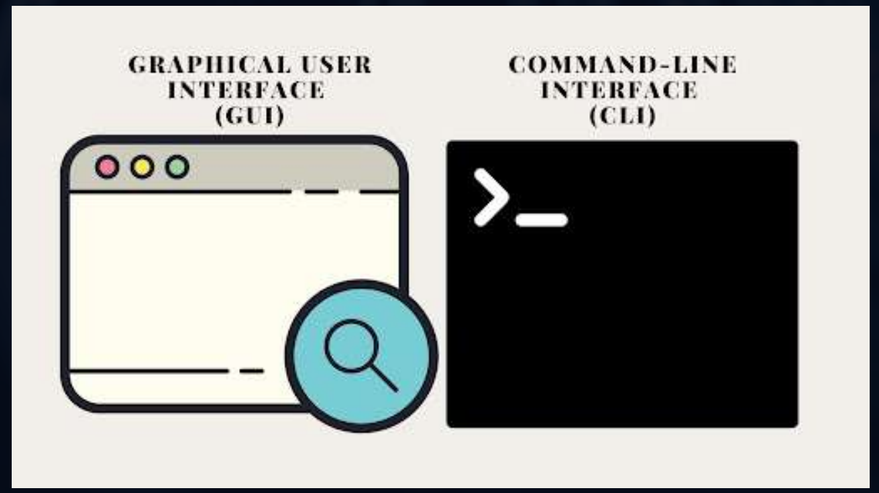
INTRODUCIÓN A UNIX

DANNA MÁRQUEZ
FERNANDO ROMERO

EJERCICIO

TERMINAL Y SHELL

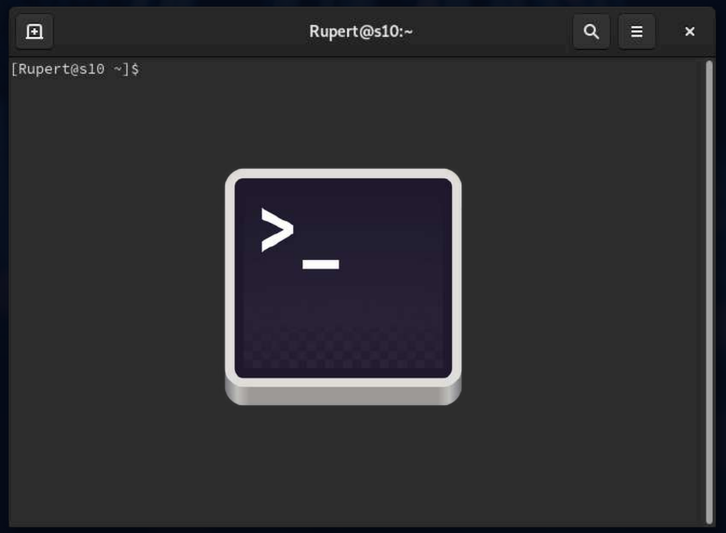
Originalmente una **terminal** o **consola** era un dispositivo que permitia interactuar entre una o más personas con las computadoras antiguas a través de una **interfaz de línea de comandos**.
En su auge, representaron una nueva etapa para la computación, facilitando las interacciones y permitiendo el modelado de aplicaciones más complejas a partir de estas.

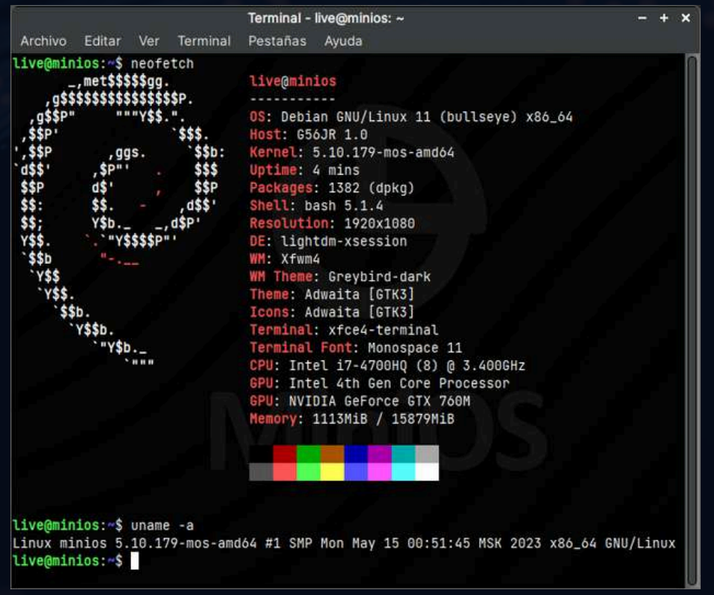


TERMINAL Y SHELL

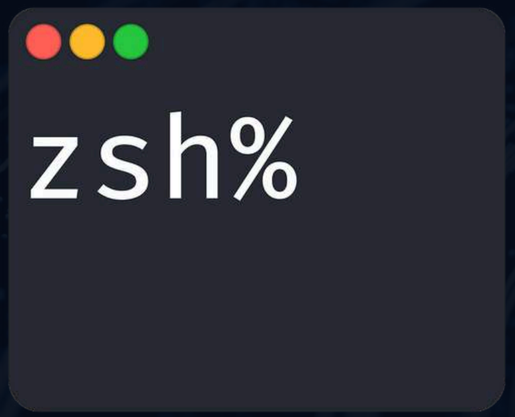
Aunque hoy en día ya no se ven estos dispositivos, ya desplazados por los monitores modernos y las interfaces gráficas de usuario, si poseemos **aplicaciones** integradas en el sistema formalmente denominadas como **emuladores de terminal**, común e imprecisamente denominados también como **terminales**.

El **emulador de terminal** es aquella aplicación que abres cuando deseas utilizar la línea de comandos para interactuar con tu sistema, como instalar aplicaciones





TERMINAL Y SHELL



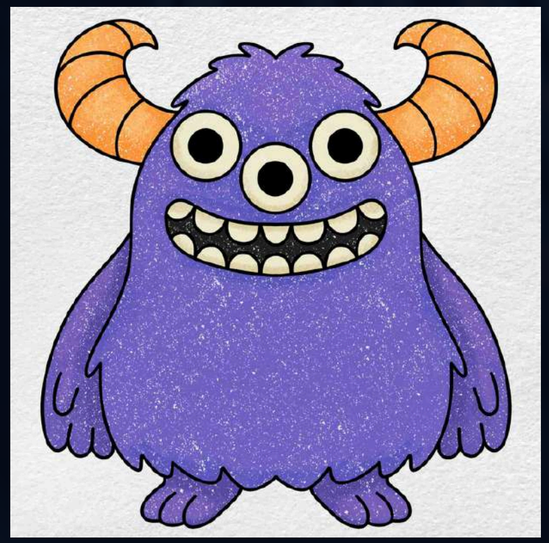
Otra parte importante que desde siempre ha acompañado a las **terminales** tanto físicas como emuladas, son las **shell**.

Una **shell** es el intérprete de comandos que corre **dentro** de la terminal, recibiendo las instrucciones del usuario, procesandolas, ejecutandolas y devolviendo los resultados a la terminal para que pueda representarlo en pantalla.

Existen varios programas **shell**, los más conocidos como **sh**, **bash**, **zsh**, **fish** y otros no tanto como **ksh**, **csh**, etc.

TERMINAL Y SHELL

Terminal



Shell

bash

TERMINAL Y SHELL

Archivos de configuración:
En cuanto a las *Shells*, éstas poseen archivos de configuración especiales que definen muchas características, ejecuciones y declaraciones que deseamos que se realicen al abrir el intérprete.

Los archivos más típicos son **.bashrc** y **.zshrc** para **Bash** y **Zsh** respectivamente. *Fish* posee un archivo de configuración distinto.

VARIABLES

En **Bash**, **Zsh** y practiacmente cualquier tipo de *Shell*, podemos declarar variables muy similar a como lo haríamos en cualquier lenguaje de programación, con el objetivo de facilitar la repetición y modularidad de instrucciones en la terminal.

Declaración

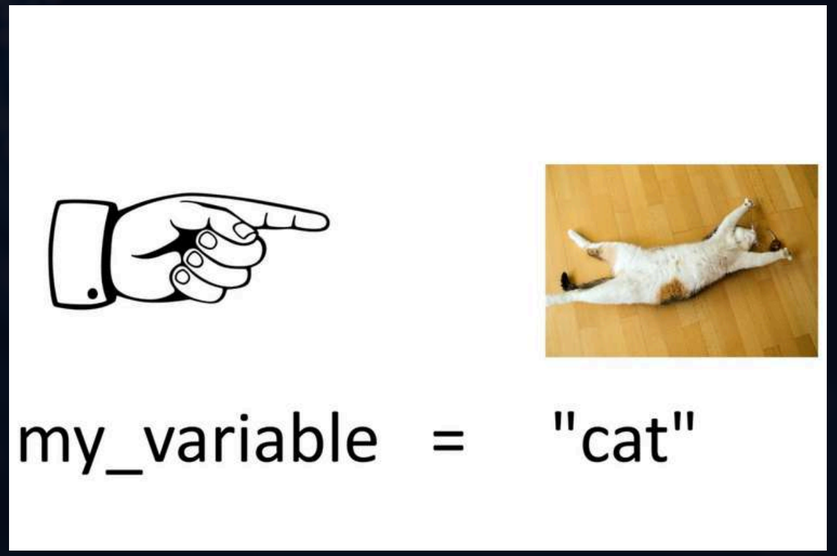
NAME=value

Uso

$NAME
echo **$NAME**
ls **$NAME**
touch **$NAME**
grep **$NAME** file.txt

Para la declaración es importante **no dejar espacios en blanco** pues la *Shell* se confunde.

Para utilizar estas variables no hay ninguna restricción, podemos utilizarlas como argumentos, banderas o hasta comandos.



VARIABLES DE ENTORNO

Las **variables de entorno** son variables **especiales** declaradas tanto en el sistema como en la *Shell*, y almacenan datos relevantes para algunos procesos y comandos.

Insistimos que, incluso fuera de la terminal, **se encuentran disponibles para cualquier proceso del sistema operativo**.

Algunos ejemplos importantes son:

- $HOME
- $EDITOR
- $SHELL
- $TERM
- $PWD
- **$?**
- **$PATH**

Los **comandos** de terminal relacionados a **variables de entorno** son:

- **env** : Lista todas las variables de entorno presentes
- **export NAME=value** : Declara una variable de entorno
- **unset** : Elimina una variable de entorno

$PATH

La variable de entorno **$PATH** define todos los directorios donde el intérprete busca los **archivos binarios** de los **comandos** que ejecutamos en la terminal.

Esto lo hace una de las variables más importantes del sistema pues brinda practicidad y comodidad al ahorrarnos escribir la ruta completa del archivo.

Comandos:

- **whereis**: Busca archivos binarios en carpetas típicas
- **which**: Busca binarios **únicamente** en las carpetas presentes en la variable $PATH
- **export PATH=”/ruta/carpeta: $PATH”**: Agregar una carpeta a $PATH

$?

Los códigos de respuesta más comunes son:

- **0**: Comando exitoso
- **1**: Comando fallido
- **2**: Sintaxis del comando incorrecta
- **126**: Intento ejecutarse un archivo sin permisos de ejecución
- **127**: Comando no encontrado
- **130**: Comando detenido por el usuario (CTRL-C)
- **131-255**: Comando detenido por el sistema (distintos casos)

La variable de entorno **$?** almacena los códigos de estado al último **comando ejecutado**.

Cuando ejecutamos un comando, puede ser que falle, que tenga éxito, que sea detenido por otro proceso, que sea detenido por nosotros, entre muchas cosas más.

Para identificar cual fue el estado o respuesta del comando se le asigna un código numérico a esta variable.

ALIAS

Podemos declarar un nombre como **alias** a otro comando, ya sea para facilitar o simplificar su uso.
Estos pueden incluir tanto banderas como argumentos ya dentro del alias, pero también reciben todos los argumentos que se les pase después.

Ejemplos:

ls == ls --color=tty
ll == ls -lh

Comandos:

- **alias** : Lista todos los alias del sistema
- **alias name='command --flag arg'**: Declara un nuevo alias
- **unalias name**: Elimina dicho alias

ATAJOS DE TECLADO

Atajo	Función
CTRL-L	Limpia la pantalla
CTRL-D	Sale de la shell actual
CTRL-Z	Suspende el proceso actual
CTRL-C	Mata el proceso actual
CTRL-H	Funciona igual que la tecla *backspace* o borrar
CTRL-A	Para ir al inicio de la línea
CTRL-W	Elimina la palabra antes del cursor
CTRL-U	Elimina desde el principio de la línea hasta la posición del cursor (o elimina toda la línea)
CTRL-E	Para ir al final de la línea
Tab	Autocompleta archivos directorios y binarios

MÁS COMANDOS

- **tree**: Lista en forma de árbol los archivos de un directorio
- **sort**: Ordena alfabéticamente el contenido de un archivo
- **uniq**: Elimina todas las lineas repetidas **adyacentes (!!)** de un archivo
- **wc**: Imprime la cuenta de líneas, palabras y bytes de un archivo
- **shuf**: Aleatoriza el orden de las líneas de un archivo
- **nano**: Editor de texto en terminal convencional
- **vim**: Editor de texto en terminal con atajos optimizados
- **bash | sh | zsh | fish**: Abre la terminal indicada

COMO USAR LINUX CON REDES?

¿Qué son?
Son sistemas de comunicación que permiten la conexión e intercambio de datos entre computadoras y otros dispositivos

COMANDOS IMPORTANTES

- **ping:** Para comprobar si un dispositivo (servidor, computadora, etc.) está accesible en la red.
- **nslookup: P**ara obtener información del DNS
- **traceroute:** Muestra la ruta que siguen los paquetes hasta llegar a un destino
- **curl:** Se usa para transferir datos desde o hacia un servidor mediante diferentes protocolos como HTTP, FTP y más.
- **wget:** Para descargar archivos de Internet de forma sencilla**.**
- **nc/ncat:** Herramienta para enviar y recibir datos a través de la red

CONTINUARÁ...