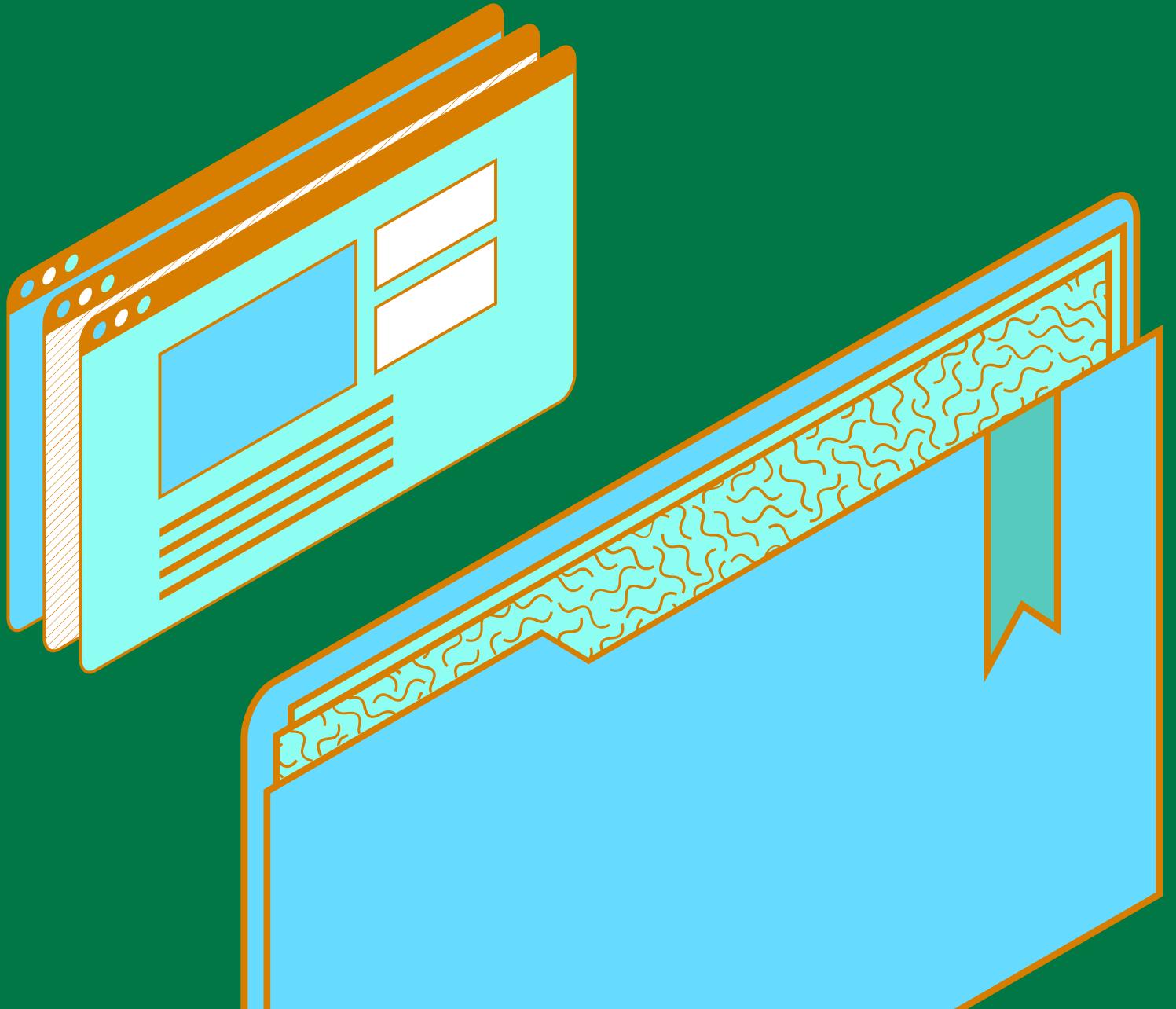


Sistemas operativos

UNIX Y SU PAPEL EN LA HISTORIA DE LOS SISTEMAS OPERATIVOS

Por Acosta Alan y Rubio Angel

ÍNDICE



01. Introducción

02. Contexto histórico

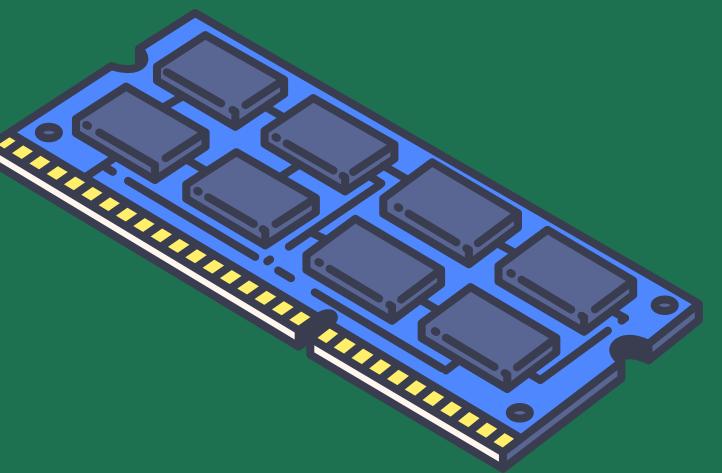
03. Arquitectura de UNIX

04. Sistema de archivos

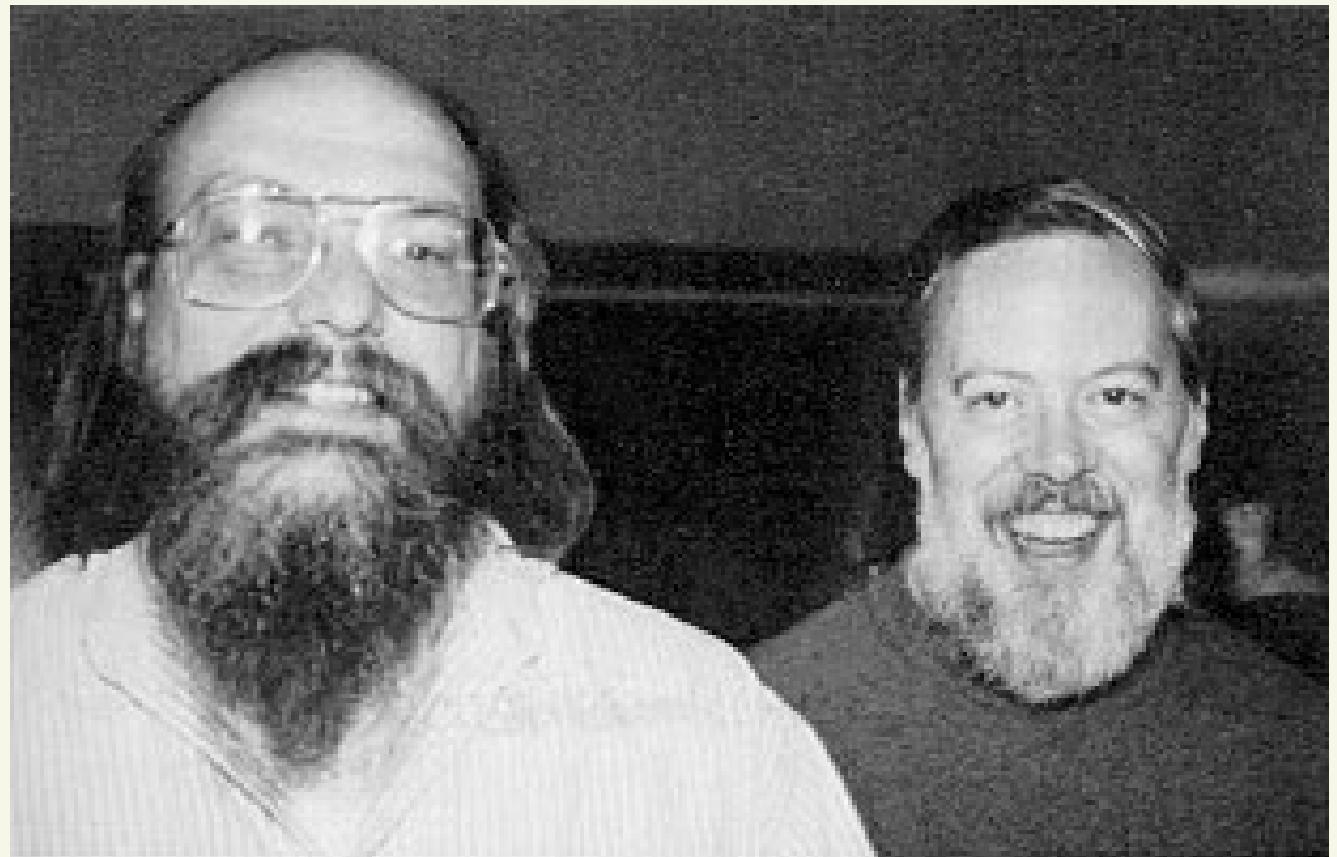
05. Comandos Basicos y importantes

06. El legado de Unix

Introducción



Unix es un sistema operativo multitarea y multiusuario que surgió en 1969 en los laboratorios Bell (Bell Labs), creado por Ken Thompson y Dennis Ritchie.



“¿Por qué hablar de UNIX hoy en día?”



"Unix is simple. It just takes a genius to understand its simplicity."

— Dennis Ritchie

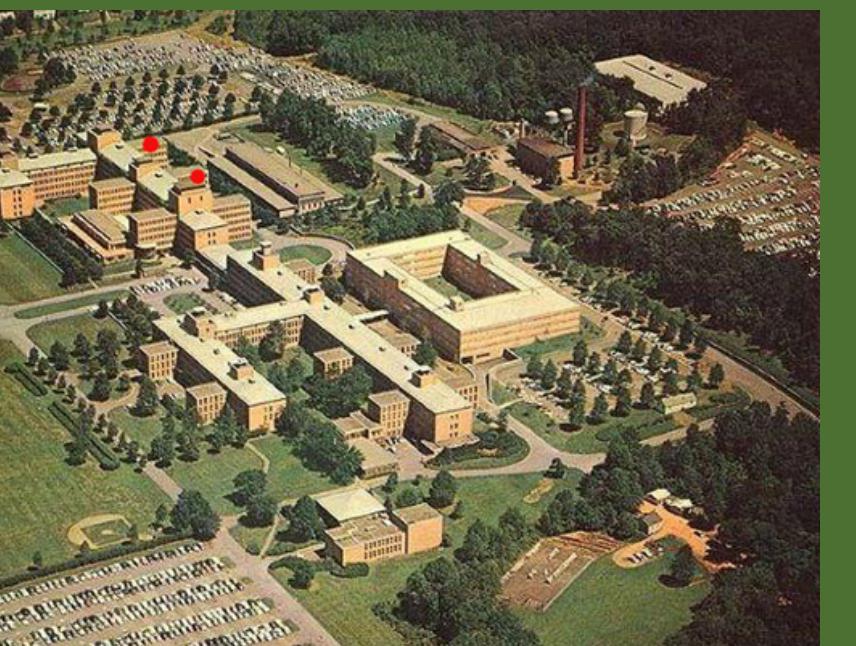
Contexto histórico



El inicio en los laboratorios Bell

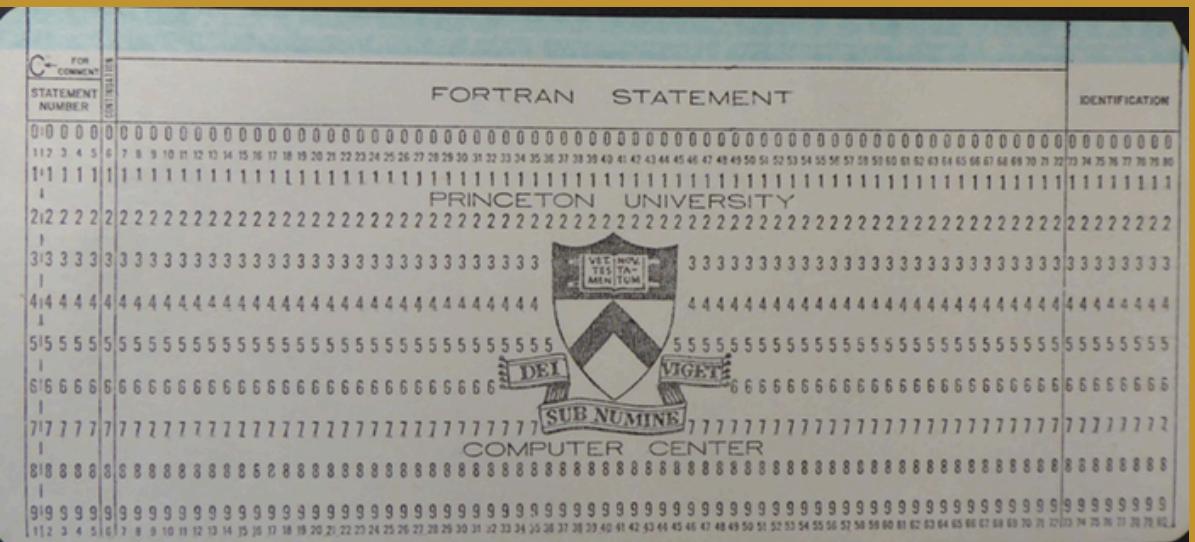


Complejo de Murray Hillen 1961



La falta de uniformidad entre los sistemas y el uso de lenguajes de bajo nivel incompatibles dificultaron enormemente el progreso

“PROCESAMIENTO POR LOTES”



Multics

Instituto de Tecnología de Massachusetts
reclutó a otras dos organizaciones:

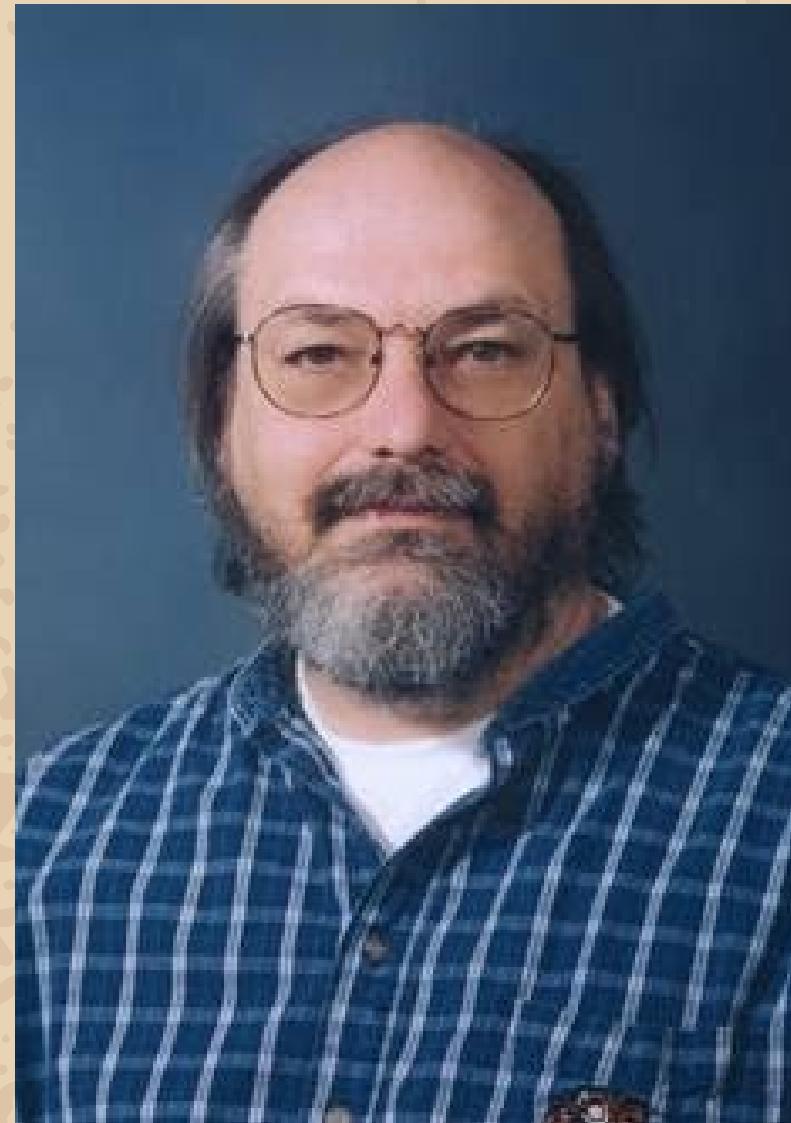


La frase "sobre-ingeniería" aparece en varias descripciones, y Sam Morgan la describió como "un intento de trepar a demasiados árboles a la vez"

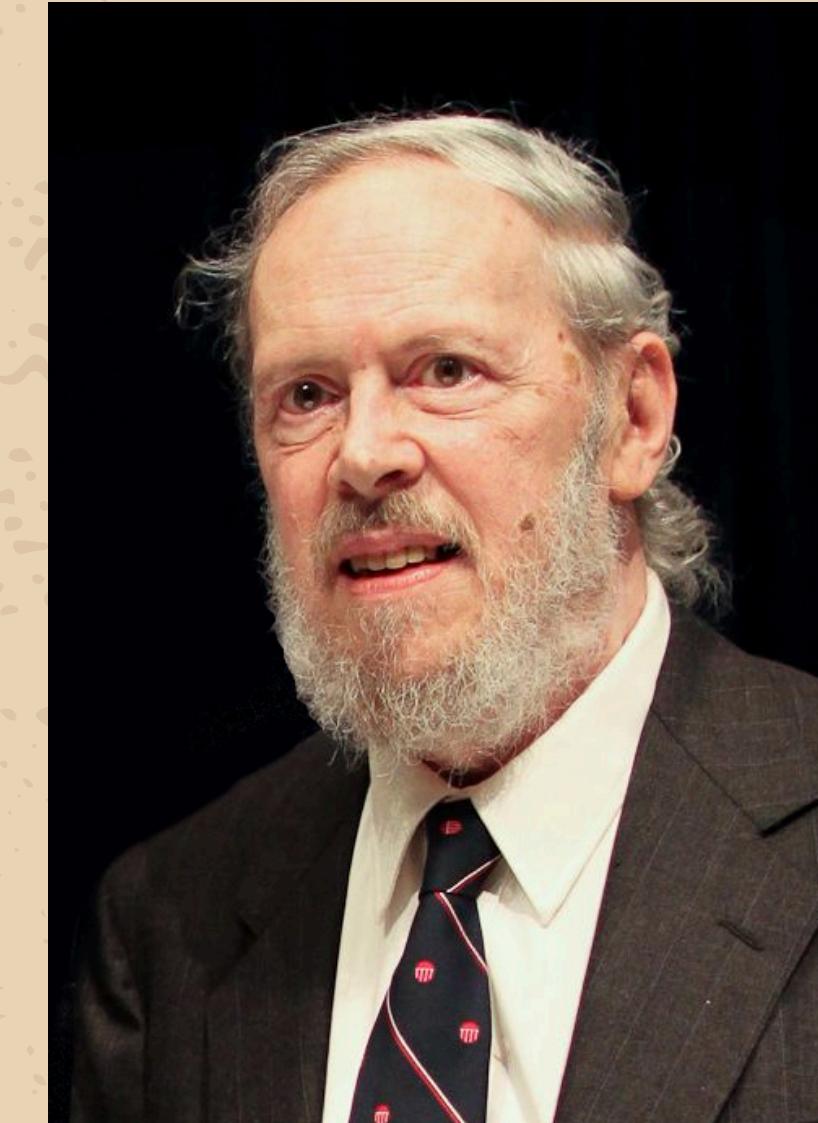


Ken Thompson estaba trabajando en Multics, Debido a la complejidad del proyecto y los problemas de integración, se unió con Dennis Ritchie "crear un sistema más simple y eficiente"

Ken Thompson y Dennis Ritchie en Bell Labs



"Ken Thompson"
Comenzó a trabajar en
Bell Labs en 1966,



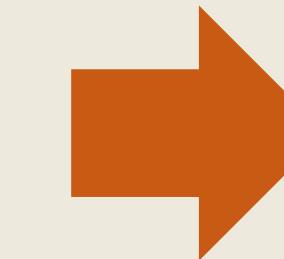
Dennis Ritchie se unió formalmente a
Bell Labs en 1967 como parte del
Computing Science Research Center

UNIX en un PDP-7 y PDP-11



¿Por qué el PDP-7?

- Tenía solo 8 KB de memoria principal (unas 8 mil palabras de 18 bits).
- Era una máquina barata, ya disponible, que no requería aprobación presupuestaria para usar.
- Su hardware era lo suficientemente flexible como para probar ideas experimentales, como sistemas de archivos, multitarea básica y llamadas al sistema.



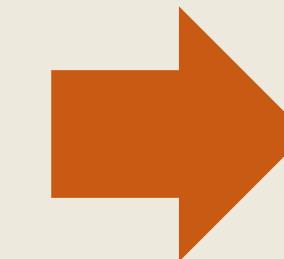
El primer Unix en el PDP-7 fue escrito completamente en lenguaje ensamblador, ya que no existía aún el lenguaje C. El sistema incluía:

¿Por qué el PDP-11?



El sistema se trasladó a una nueva computadora, que ya tenía más capacidad y mejor soporte para multitarea.

la PDP-11 fue la máquina más importante utilizada en los laboratorios Bell para el desarrollo de Unix



Con 24 KB de memoria y discos más grandes, el PDP-11 ofreció un mejor rendimiento

“Unix fue creado con cero presupuesto, un equipo olvidado, y la necesidad de jugar un videojuego.”
— Parafraseando a Brian Kernighan

ARQUITECTURA DE UNIX(UNA REEVOLUCION)

KERNEL, SHELL Y SISTEMA DE ARCHIVOS

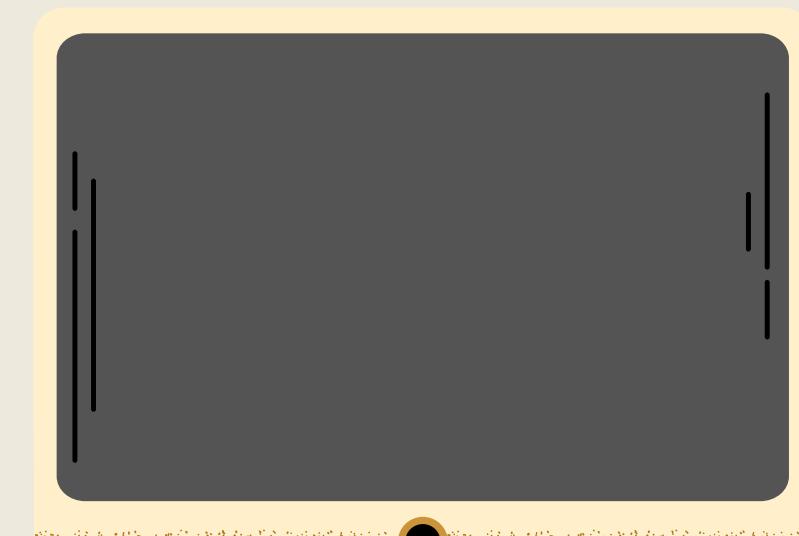
KERNEL Y SHELL

Shell y el Kernel trabajan en conjunto

KERNEL

Es el núcleo del sistema.

Se encarga de gestionar los recursos de hardware, como la memoria, el procesador, el sistema de archivos y los dispositivos.



SHELL

Interprete de linea de comando

Es la interfaz entre el usuario y el kernel. Interpreta los comandos introducidos por el usuario y los traduce en llamadas al sistema.



Para ilustrar la interacción entre el shell y el kernel, supongamos que un usuario escribe **rm myfile** (lo que elimina el archivo **myfile**).

rm myfile

PASO 1: SE ACCDEDE A LA SHEL MEDIANTE LA INTERFAZ

Cuando el usuario escribe el comando **rm myfile** se crea un script que se ejecuta en la terminal(considerar que depende del sistema)

La terminal accede a las aplicaciones del sistemas (comandos /programas)

El shell busca en el sistema de archivos para encontrar la ruta de ese programa

PASO 2: LLAMADA AL SISTEMA (SYSTEM CALL)

Una llamada al sistema es una interfaz que permite a los programas de usuario (como el shell) solicitar servicios del núcleo



Este ejemplo muestra cómo un comando escrito por el usuario en la terminal de UNIX pasa por varias capas del sistema antes de ejecutarse, desde el shell hasta el núcleo del sistema operativo.

PASO 3: EL NÚCLEO EJECUTA EL COMANDO

Todo comando hará uso del kernel, donde este le proporciona recursos al comando rm

Asigna tiempo de CPU para que el programa rm se ejecute.

Accede al sistema de archivos para buscar el archivo myfile y luego lo elimina.

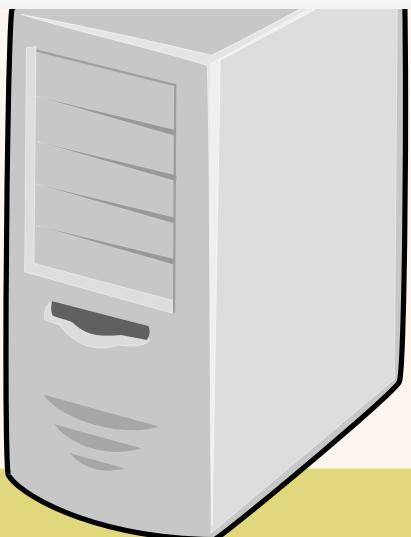
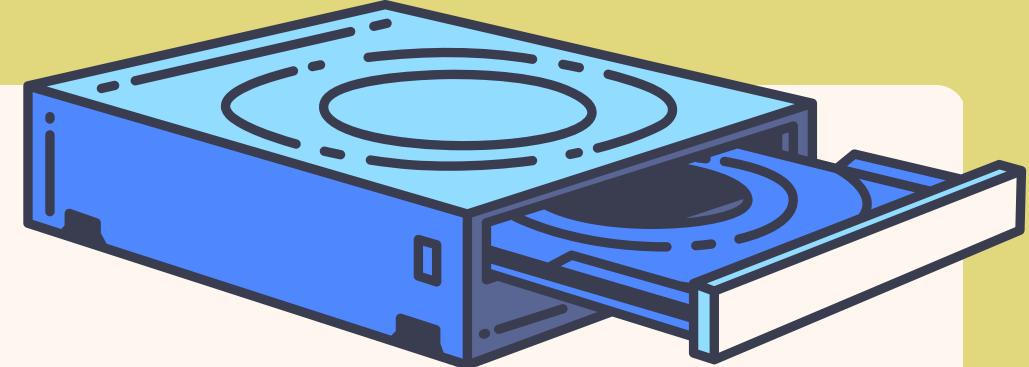
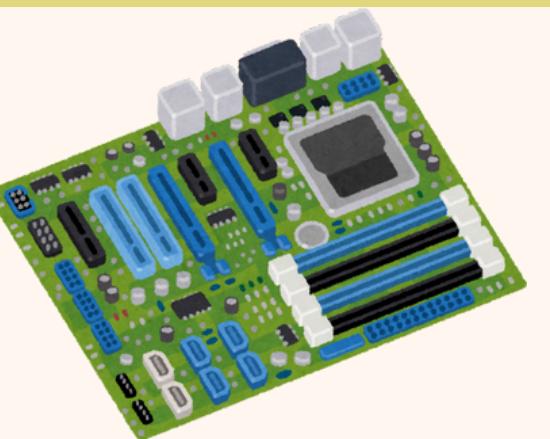
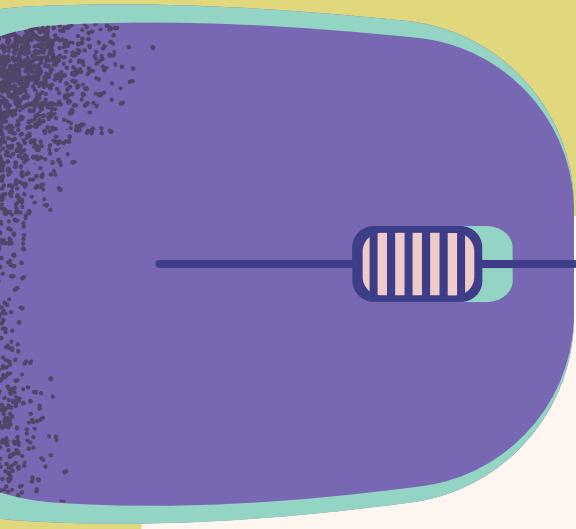
El núcleo también gestiona la memoria utilizada

PASO 4: FINALIZACIÓN DEL COMANDO

Una vez que el programa rm ha terminado de eliminar el archivo myfile, el proceso de eliminación se completa y el control regresa al shell para mostrarlo.

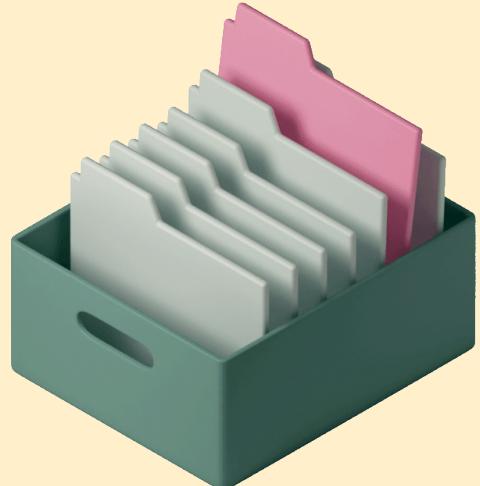
SHELL Y KERNELL

```
rubio@MacBook-Air-de-Jose Expo % touch prueba.txt
rubio@MacBook-Air-de-Jose Expo % echo "Soy el shell"
Soy el shell
rubio@MacBook-Air-de-Jose Expo % Echo $$SHELL
/bin/zsh
rubio@MacBook-Air-de-Jose Expo % ps
  PID TTY          TIME CMD
    554 ttys000  0:01.15 -zsh
rubio@MacBook-Air-de-Jose Expo % rm prueba.txt
rubio@MacBook-Air-de-Jose Expo % uname -a
Darwin MacBookAir.lan 24.2.0 Darwin Kernel Version 24.2.0: Fri Dec  6 18:48:14 PST 2024; root:xnu-11215.61.5~2/RELEASE_ARM64_T8103
  arm64
rubio@MacBook-Air-de-Jose Expo % █
```

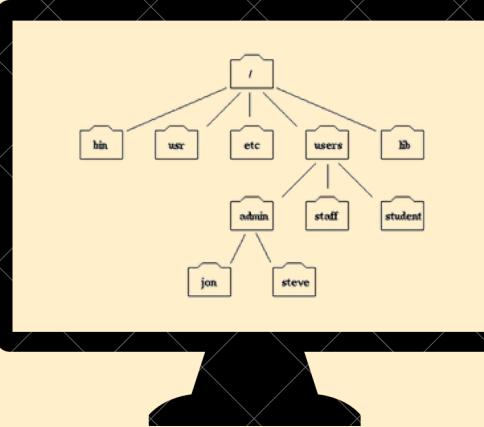


SISTEMA DE ARCHIVOS

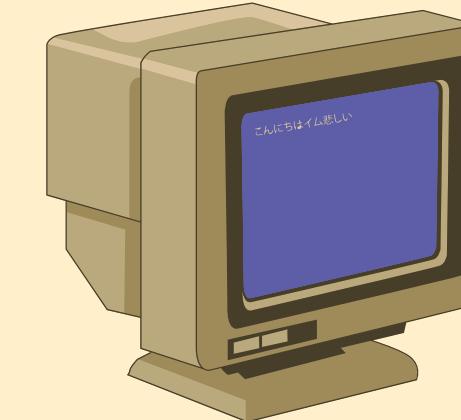
“En Unix, todo es un archivo”



**ARCHIVOS
REGULARES:**



DIRECTORIOS



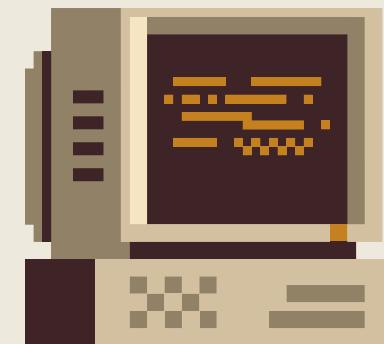
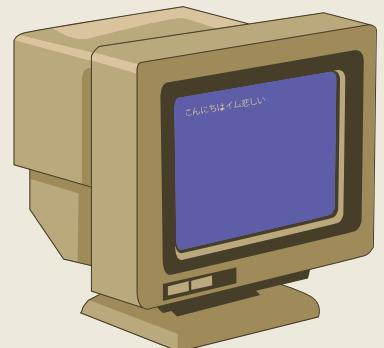
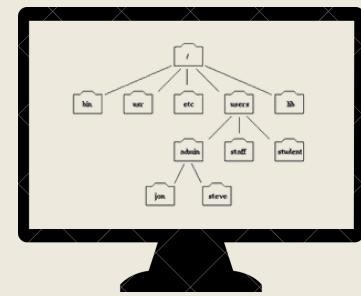
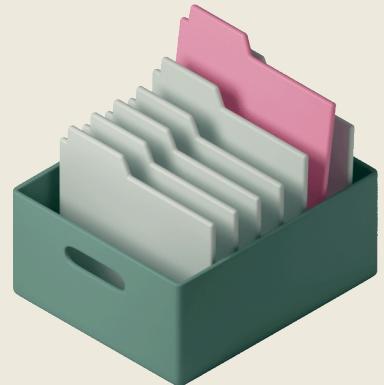
**ARCHIVOS DE
DISPOSITIVO:**



**ENLACES
SIMBÓLICOS**



SISTEMA DE ARCHIVOS



ARCHIVOS REGULARES:
CONTIENEN DATOS, TEXTO O CÓDIGO EJECUTABLE.



DIRECTORIOS:
SON ARCHIVOS ESPECIALES QUE ALMACENAN REFERENCIAS A OTROS ARCHIVOS Y DIRECTORIOS.
SON LA BASE DEL SISTEMA DE ARCHIVOS JERÁRQUICO DE UNIX.



ARCHIVOS DE DISPOSITIVO:
REPRESENTAN DISPOSITIVOS FÍSICOS COMO DISCOS, IMPRESORAS O TERMINALES. SE DIVIDEN EN BLOQUES (ALMACENAMIENTO) Y CARACTERES (ENTRADA/SALIDA).



ENLACES SIMBÓLICOS:
SON ACCESOS DIRECTOS QUE APUNTAN A OTRO ARCHIVO.

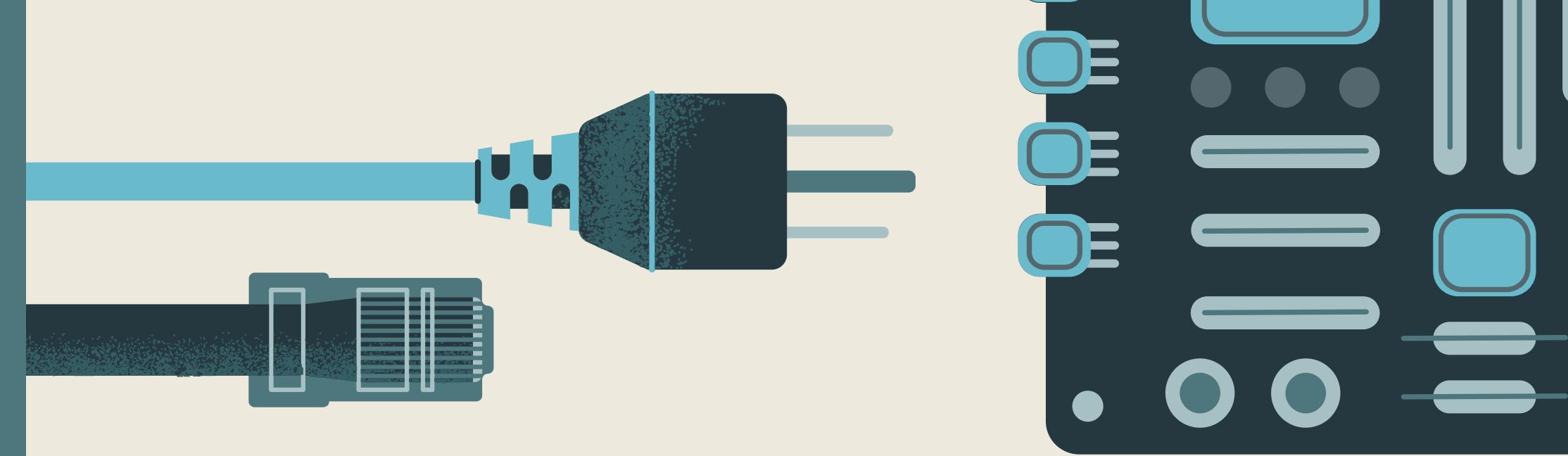
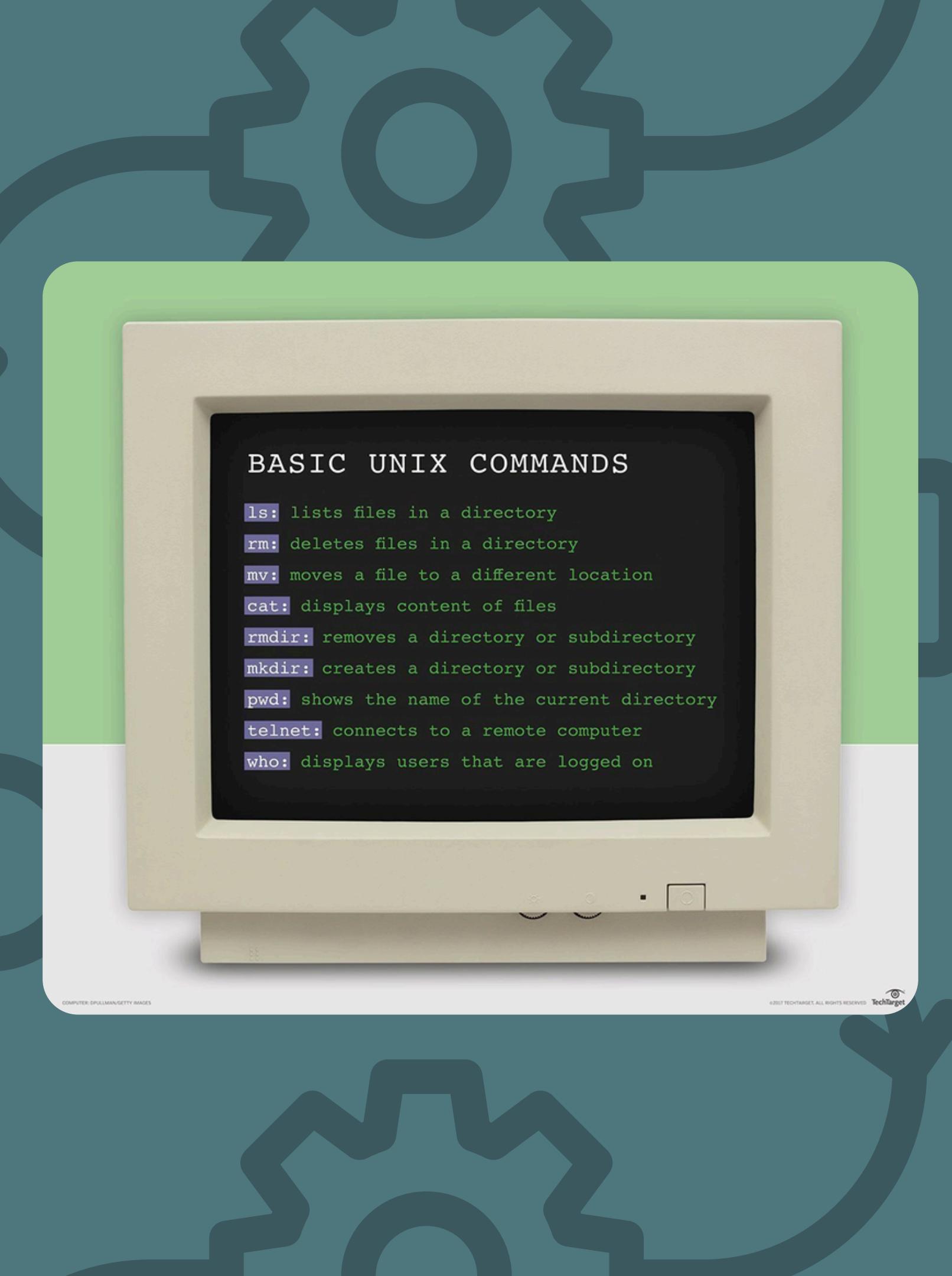


ESPECIALES

Representan rutinas del kernel que proporcionan acceso a diversos dispositivos del sistema;

Bloque
Liga Simbolica
Socket
Pipe

- **SÍMBOLO SIGNIFICADO**
 - Ordinario
 - d Directorio
 - c Carácter
 - b Bloque
 - I Liga simbólica
 - s Socket
 - p Pipe



COMANDOS BASICOS DE UNIX

Unix se caracteriza por ofrecer comandos concisos, fáciles de recordar y orientados al usuario, los cuales permiten acceder rápidamente a información relevante del sistema.

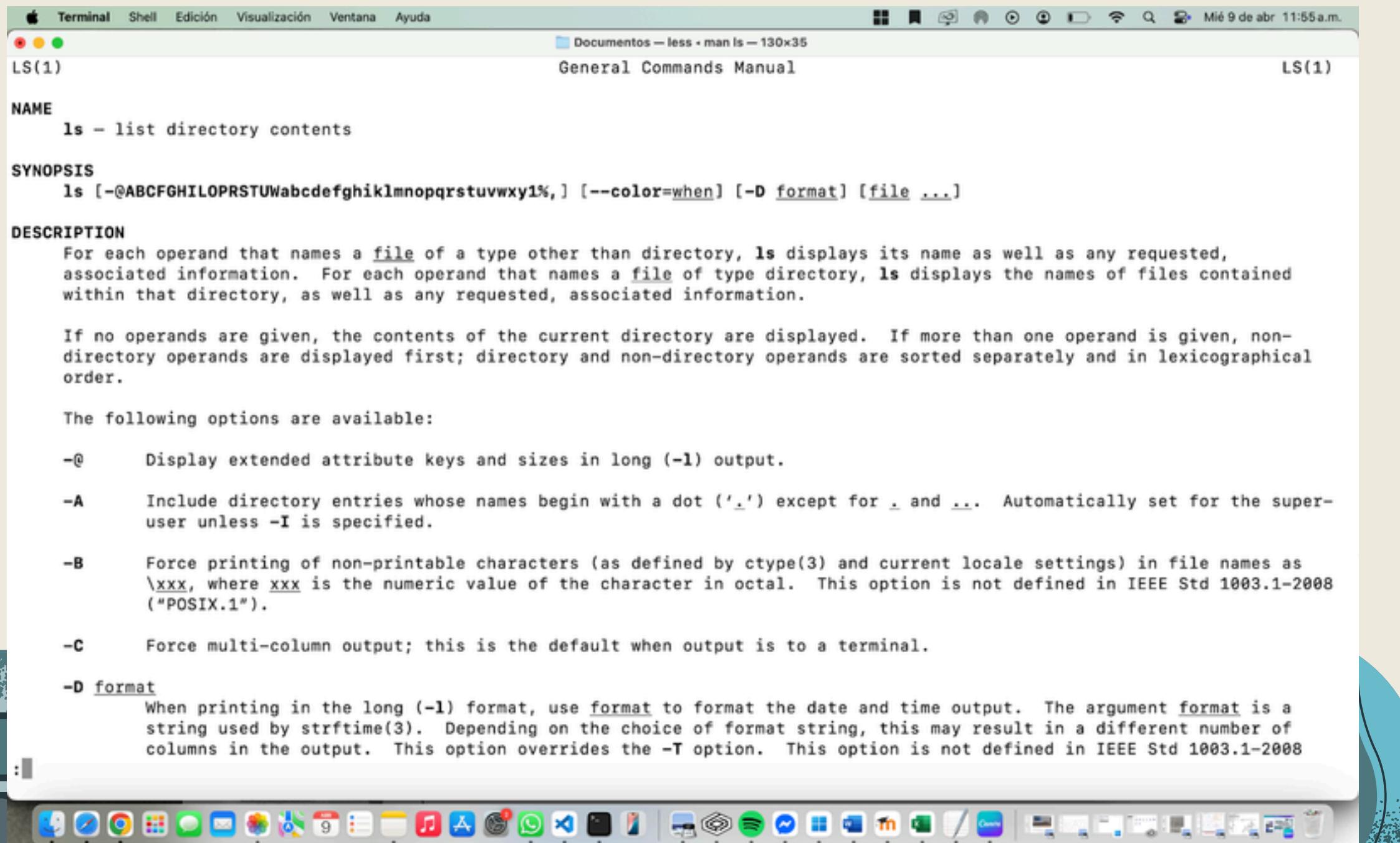
BASIC UNIX COMMANDS

```
ls: lists files in a directory
rm: deletes files in a directory
mv: moves a file to a different location
cat: displays content of files
rmdir: removes a directory or subdirectory
mkdir: creates a directory or subdirectory
pwd: shows the name of the current directory
telnet: connects to a remote computer
who: displays users that are logged on
```

COMANDO “MAN”

El comando man sustituyó la consulta tradicional al “Programmer’s Manual.”

Se utiliza escribiendo “man” seguido del comando que se desea consultar, por ejemplo: man ls.



The screenshot shows a Mac OS X desktop with a terminal window open. The window title is "Documentos - less - man ls - 130x35". The content of the window is the man page for the "ls" command. It includes sections for NAME, SYNOPSIS, and DESCRIPTION, along with details about options like -A, -B, and -D. The terminal window is part of a larger desktop environment with other application icons visible in the Dock at the bottom.

```
LS(1)
General Commands Manual
LS(1)

NAME
ls - list directory contents

SYNOPSIS
ls [-@ABCFGHIOPRSTUWabcdefghijklmnopqrstuvwxyz1%,] [--color=when] [-D format] [file ...]

DESCRIPTION
For each operand that names a file of a type other than directory, ls displays its name as well as any requested, associated information. For each operand that names a file of type directory, ls displays the names of files contained within that directory, as well as any requested, associated information.

If no operands are given, the contents of the current directory are displayed. If more than one operand is given, non-directory operands are displayed first; directory and non-directory operands are sorted separately and in lexicographical order.

The following options are available:

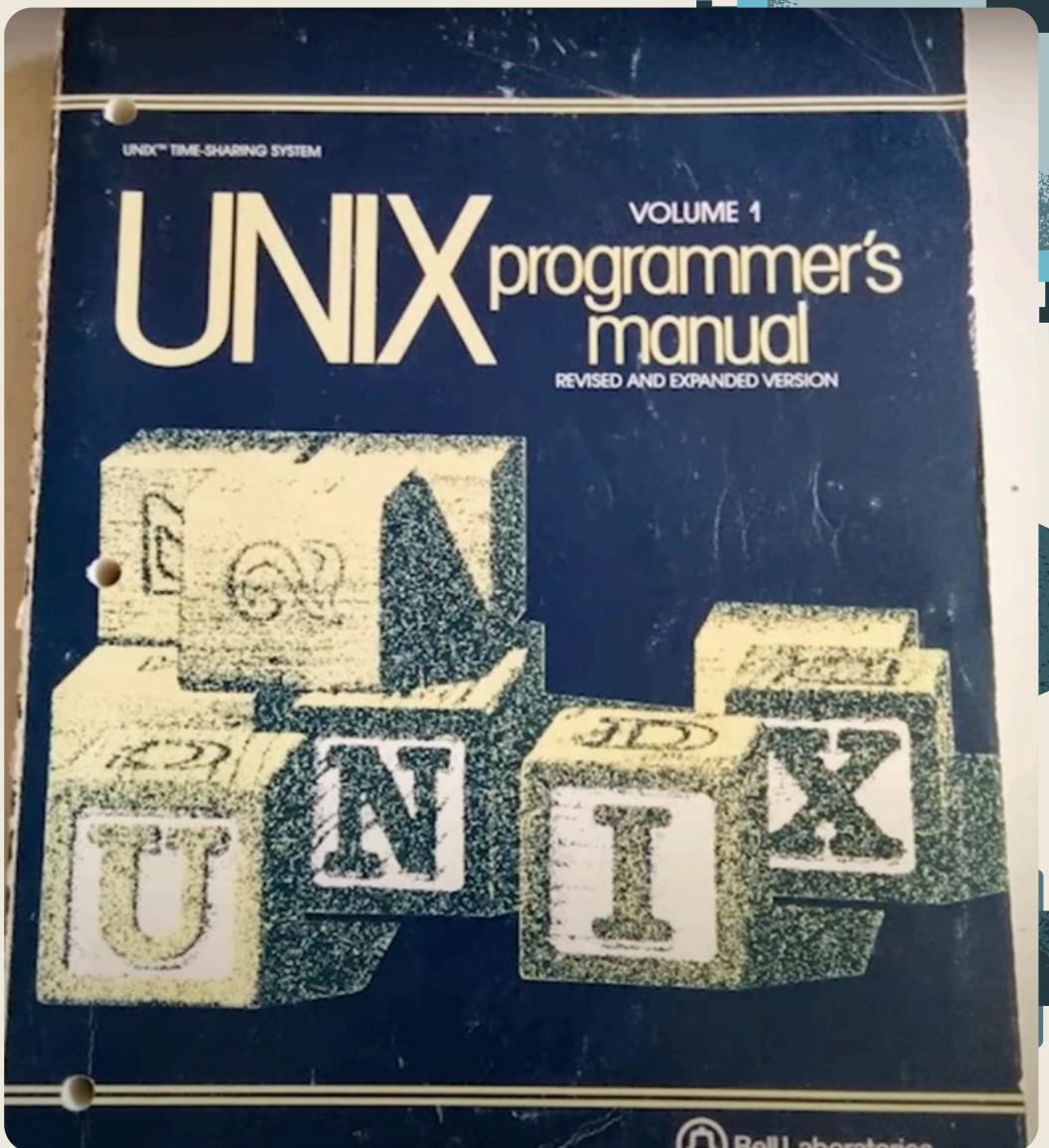
-@    Display extended attribute keys and sizes in long (-l) output.

-A    Include directory entries whose names begin with a dot ('.') except for '.' and '...'. Automatically set for the super-user unless -I is specified.

-B    Force printing of non-printable characters (as defined by ctype(3) and current locale settings) in file names as '\xxx, where xxx is the numeric value of the character in octal. This option is not defined in IEEE Std 1003.1-2008 ("POSIX.1").

-C    Force multi-column output; this is the default when output is to a terminal.

-D format
When printing in the long (-l) format, use format to format the date and time output. The argument format is a string used by strftime(3). Depending on the choice of format string, this may result in a different number of columns in the output. This option overrides the -T option. This option is not defined in IEEE Std 1003.1-2008
```



MANUAL GUIA PARA USAR UNIX

COMANDO GREP

Global / Regular expression / Print

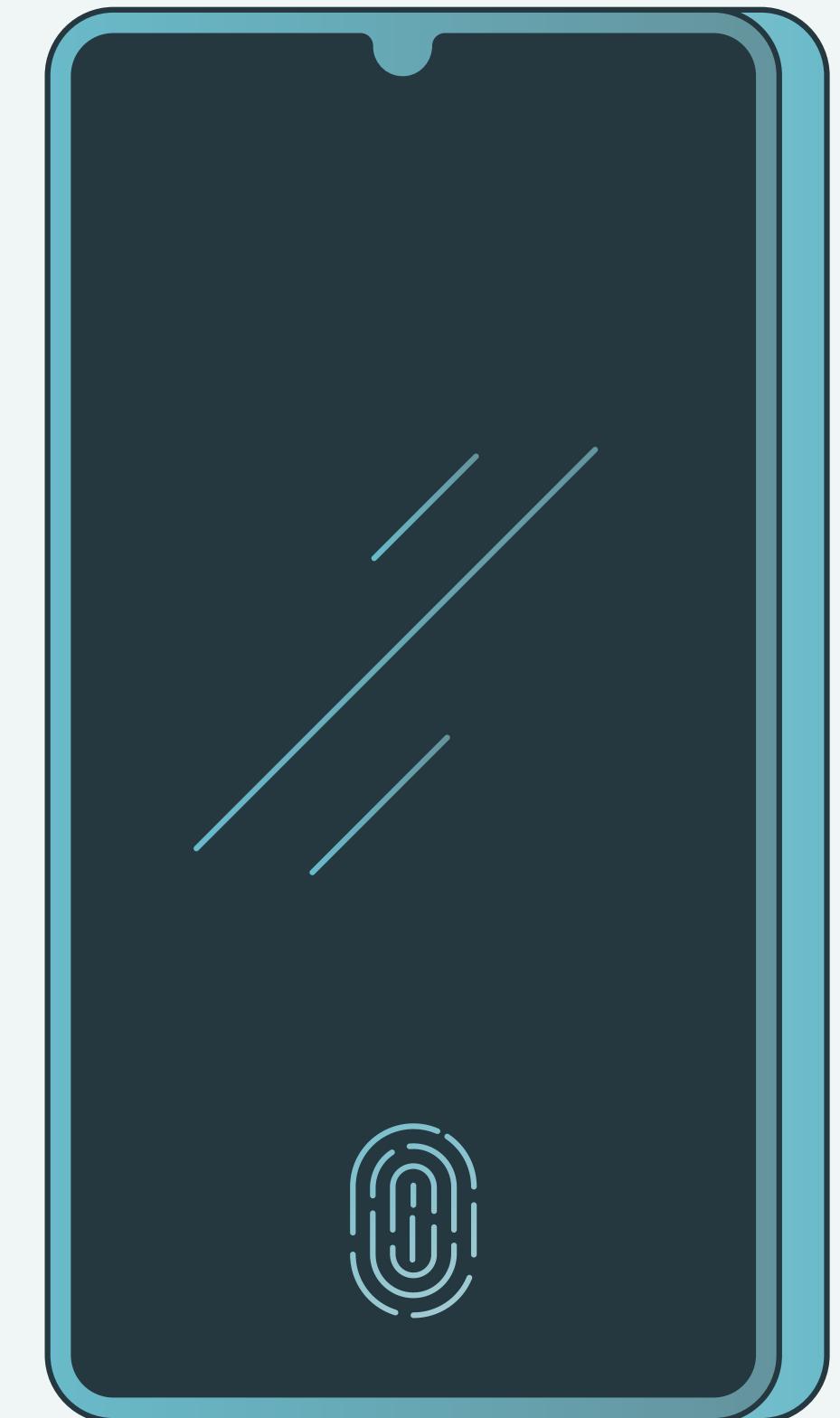
```
rubio@MacBook-Air-de-Jose Texto.txt % touch Error.txt
rubio@MacBook-Air-de-Jose Texto.txt % echo "Aqui se encuentra dos veces la palabra error (error)" > Error.txt
rubio@MacBook-Air-de-Jose Texto.txt % cat Error.txt
Aqui se encuentra dos veces la palabra error (error)
rubio@MacBook-Air-de-Jose Texto.txt % grep 'error' Error.txt
Aqui se encuentra dos veces la palabra error (error)
rubio@MacBook-Air-de-Jose Texto.txt % echo "Aqui no se encuentra la palabra" >> Error.txt
rubio@MacBook-Air-de-Jose Texto.txt % grep 'error' Error.txt
Aqui se encuentra dos veces la palabra error (error)
rubio@MacBook-Air-de-Jose Texto.txt % grep 'palabra' Error.txt
Aqui se encuentra dos veces la palabra error (error)
Aqui no se encuentra la palabra
rubio@MacBook-Air-de-Jose Texto.txt % echo "Hola" >> Error.txt
rubio@MacBook-Air-de-Jose Texto.txt % grep 'Hola' Error.txt
Hola
rubio@MacBook-Air-de-Jose Texto.txt % grep 'o' Error.txt
Aqui se encuentra dos veces la palabra error (error)
Aqui no se encuentra la palabra
Hola
rubio@MacBook-Air-de-Jose Texto.txt % grep 'H' Error.txt
Hola
rubio@MacBook-Air-de-Jose Texto.txt % grep 'h' Error.txt
rubio@MacBook-Air-de-Jose Texto.txt % grep -i 'h' Error.txt
Hola
rubio@MacBook-Air-de-Jose Texto.txt %
```



COMANDO SED

stream editor

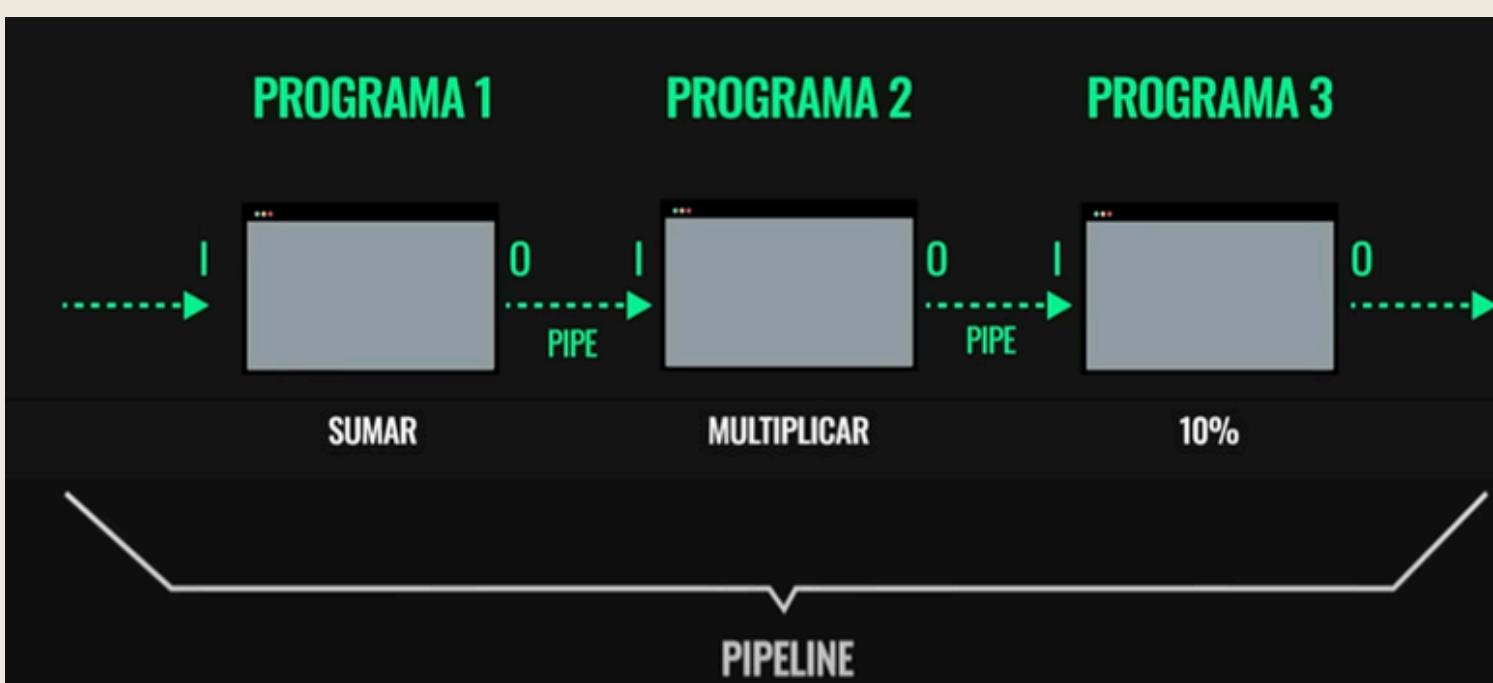
```
rubio@MacBook-Air-de-Jose Texto.txt % echo probando el comando sed  
probando el comando sed  
[rubio@MacBook-Air-de-Jose Texto.txt % sed 's/#/aqui/' Error.txt  
  
Solo remplaza la primer ocurrencia (aqui, # ,#)  
  
Remplaza todas las ocurrencias(%, %, %)  
  
[Las remplaza y lo guarda (cambio, cambio )%  
rubio@MacBook-Air-de-Jose Texto.txt % sed 's/%/aqui/g' Error.txt  
  
Solo remplaza la primer ocurrencia (#, # ,#)  
  
Remplaza todas las ocurrencias(aqui, aqui, aqui)  
  
[Las remplaza y lo guarda (cambio, cambio )%  
rubio@MacBook-Air-de-Jose Texto.txt % sed -i '' 's/cambio/Hola/g' Error.txt  
[rubio@MacBook-Air-de-Jose Texto.txt % cat Error.txt  
  
Solo remplaza la primer ocurrencia (#, # ,#)  
  
Remplaza todas las ocurrencias(%, %, %)  
  
Las remplaza y lo guarda (Hola, Hola )%  
rubio@MacBook-Air-de-Jose Texto.txt %
```



PIPES

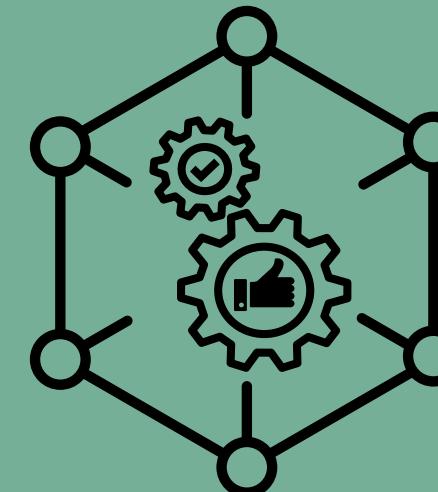
Forma de conectar la salida de un comando con la entrada de otro.

PIPELINE



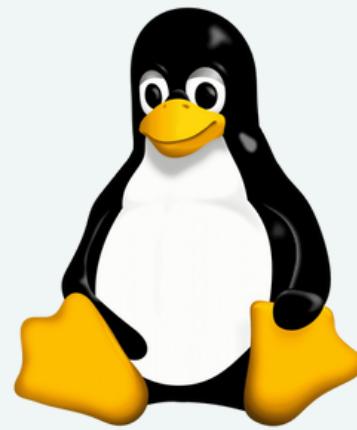
```
rubio@MacBook-Air-de-Jose Expo % cat > servidor.log <<EOF
2025-04-14 08:00 INFO conexión establecida con 192.168.1.1
2025-04-14 08:05 ERROR conexión fallida con 192.168.1.2
2025-04-14 09:10 INFO conexión establecida con 192.168.1.3
2025-04-14 10:45 ERROR conexión fallida con 192.168.1.4
2025-04-14 11:00 ERROR conexión fallida con 192.168.1.5
2025-04-13 17:30 ERROR conexión fallida con 192.168.1.6
2025-04-14 13:00 INFO conexión establecida con 192.168.1.7
2025-04-14 13:15 ERROR conexión fallida con 192.168.1.8
2024-04-14 15:30 ERROR conexión fallida con 192.168.1.9
EOF
rubio@MacBook-Air-de-Jose Expo % cat servidor.log | grep "ERR
4
rubio@MacBook-Air-de-Jose Expo %
```

```
rubio@MacBook-Air-de-Jose Expo % cat servidor.log | grep "ERROR conexión" | grep "2025-04-14" | wc -l
4
rubio@MacBook-Air-de-Jose Expo %
```



UNIX Y SU LEGADO

UNIX WAS NOT DESIGNED TO STOP ITS USERS FROM DOING STUPID THINGS, AS THAT WOULD ALSO STOP THEM FROM DOING CLEVER THINGS."



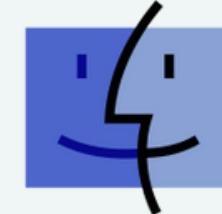
1991

Linux



1993

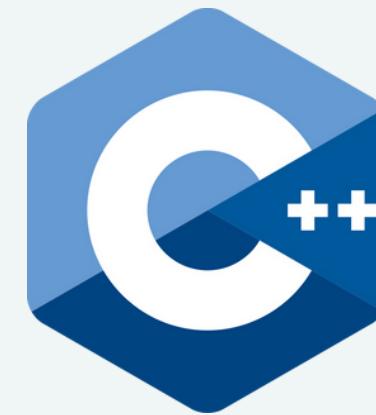
Windows NT



Mac OS

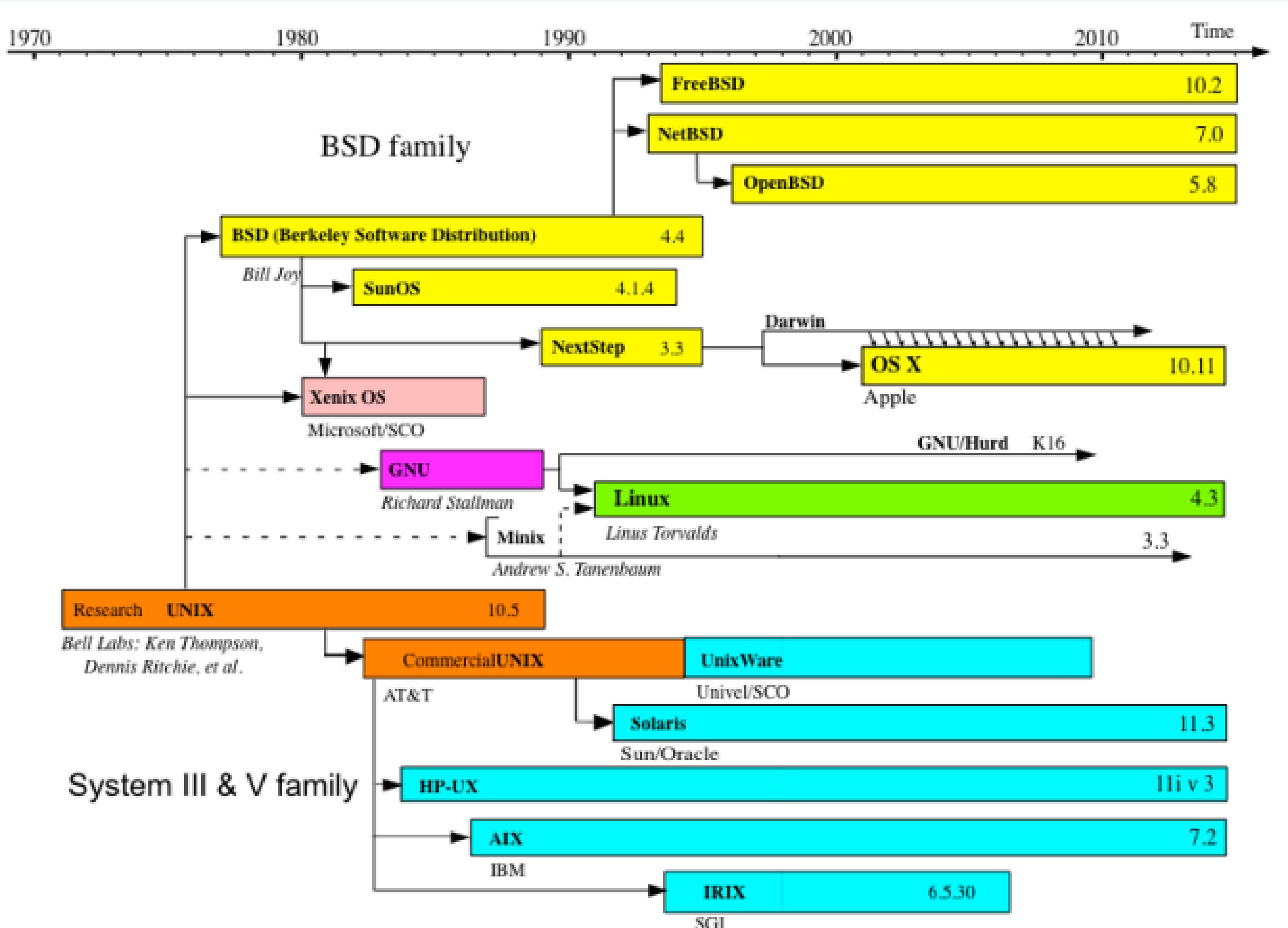
1997

MacOs



1997+

Lenguaje C y C++



CONCLUSIONES



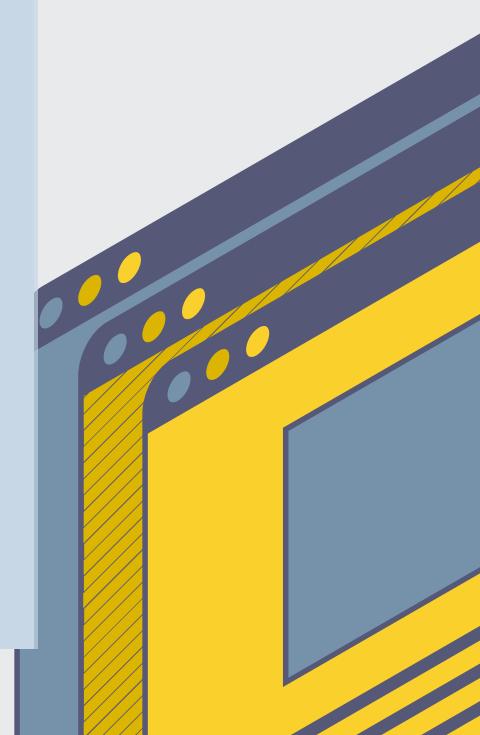
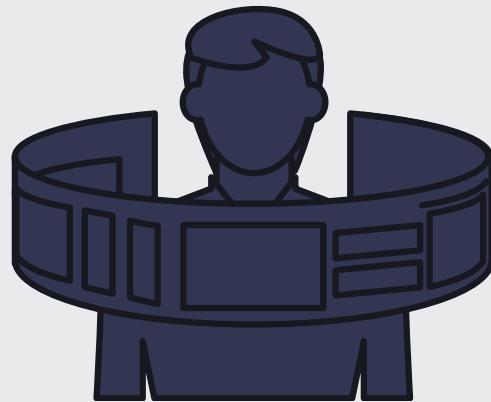
01.

UNIX marcó un antes y un después en la historia de la computación. No solo introdujo avances técnicos fundamentales —como la multitarea, el sistema de archivos jerárquico y el uso de lenguajes portables como C—, sino que definió una filosofía de diseño que ha influido en generaciones enteras de ingenieros, desarrolladores y científicos.



02.

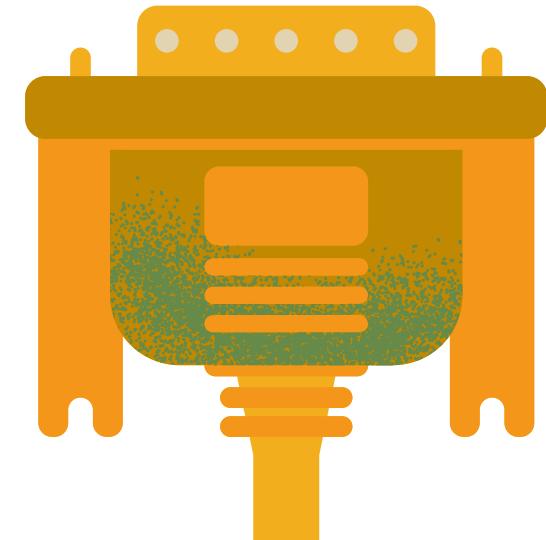
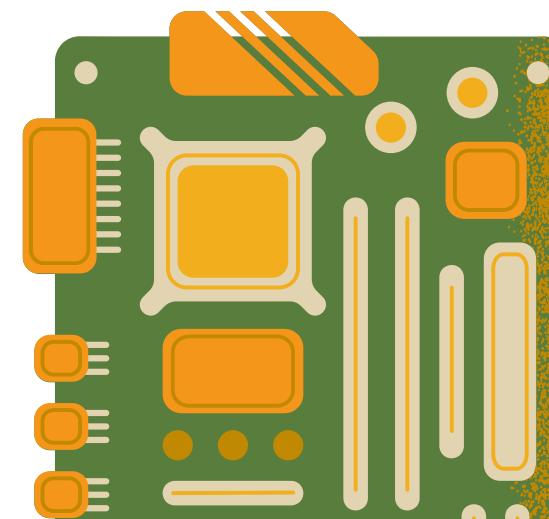
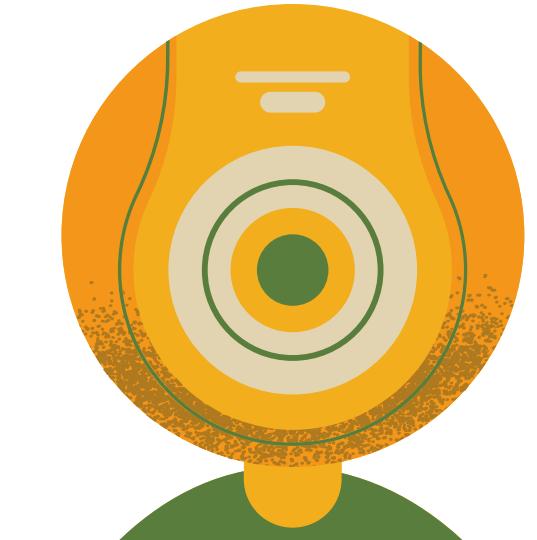
UNIX marcó un antes y un después en la historia de los sistemas operativos gracias a su diseño simple, modular y portátil, que influyó directamente o indirectamente en la creación de sistemas como Linux, macOS y Windows NT. Su filosofía de herramientas pequeñas y combinables, junto con su desarrollo en el lenguaje C, su desarrollo en el lenguaje C permitió que fuera fácilmente adaptado y compartido, lo que lo convirtió en la base de muchos otros sistemas. Más allá de su código, UNIX dejó una huella profunda en la filosofía del software: colaboración, reutilización y control total sobre la máquina. Hoy, su legado sigue vivo en casi todo lo que usamos.





Sistemas Operativos

MUCHAS GRACIAS



Fuentes de Información

Hostinger Academy. (2021, 24 de febrero). *What is UNIX? UNIX Operating System Explained [Video]*. YouTube. <https://www.youtube.com/watch?v=KuGCq7L6WaM>

Hostinger. (n.d.). *50 Most Commonly Used Linux Commands (With Examples)*. Hostinger Tutorials. <https://www.hostinger.com/tutorials/linux-commands>

UNIX / Linux tutorial for beginners. (n.d.). Retrieved April 17, 2025, from Surrey.ac.uk website: <https://info-ee.surrey.ac.uk/Teaching/Unix/>

0Lourdes Yolanda Flores Salgado, M. (n.d.). Introducción a UNIX. Retrieved April 17, 2025, from Unam.mx website:
https://triton.astroscu.unam.mx/fruiz/introunix_plan_2013.pdf