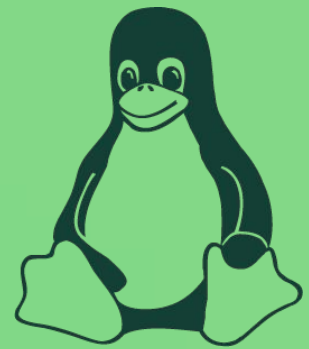


Curso de

# **Introducción a la Administración de Servidores Linux**

Enrique Devars  
@codevars







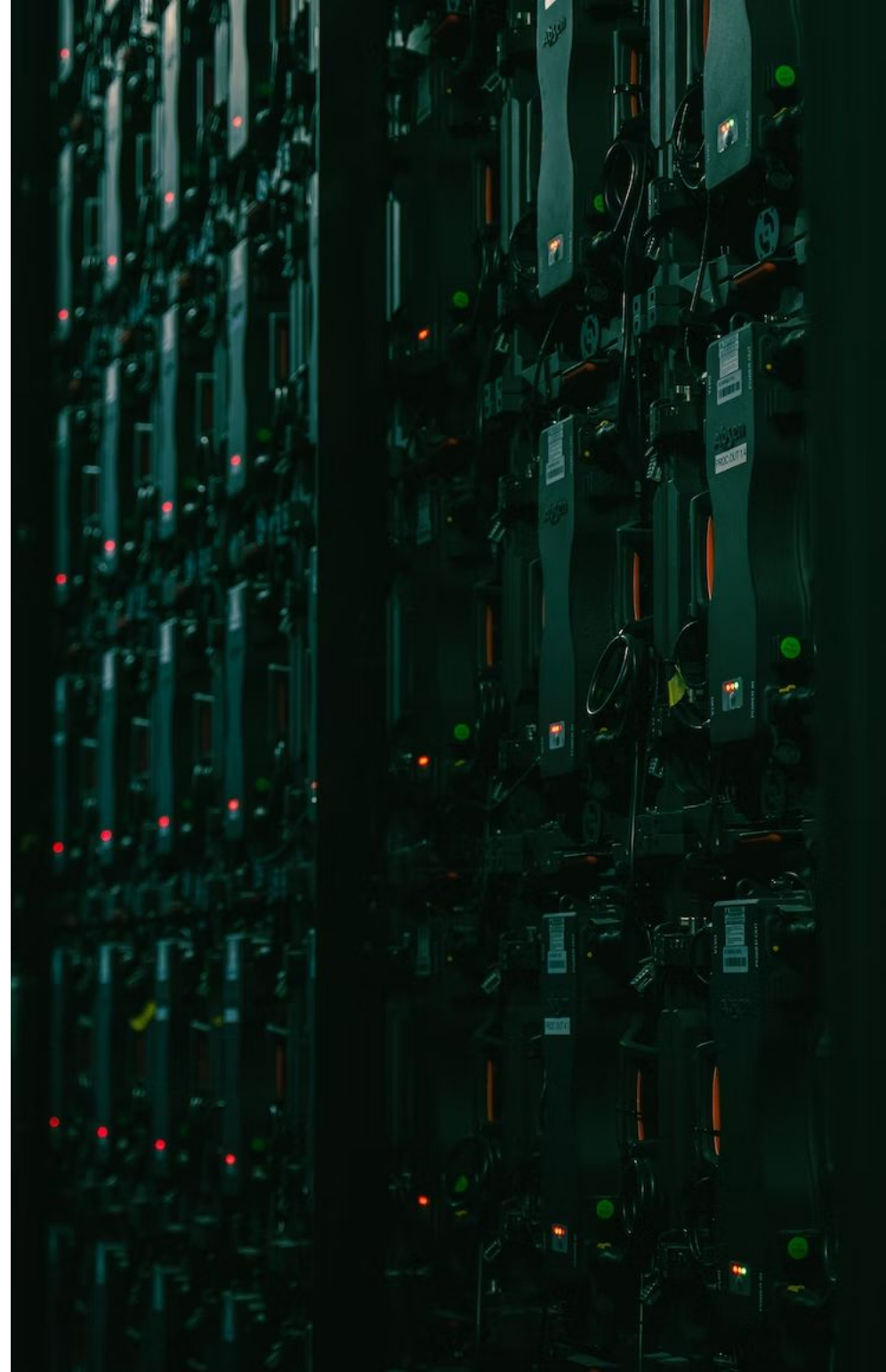


# ¿Qué necesitas para iniciar este curso?

- Conceptos básicos de ingeniería de software.
- Manejo básico de la terminal.
- Manejo de algún lenguaje de scripting como Shell o Python. (Opcional).
- Conocimiento de redes computacionales. (Opcional).

# ¿Quiénes usan Linux?

- **+80%** de mercado de servidores.
- **El 100%** del top 500 supercomputadoras.
- **90%** de los servidores en la nube.







# Objetivo del curso

- Familiarizarte con los diferentes tipos de servidores.
- Aprender los conceptos claves detrás de un sistema Linux.
- Administrar de manera básica un servidor Linux.
- Entender cómo funciona un sistema Linux por dentro.



# **Habilidades y roles de un administrador Linux**





# Habilidades clave

- Control de accesos
- Monitoreo del sistema
- Administración de recursos
- *Troubleshooting*
- Instalación y mantenimiento de software
- Creación de respaldos
- Documentación



# **Roles que puedes desempeñar**

- DevOps Engineer.

Se enfocan en los procesos y metodologías para la correcta liberación en el proceso de desarrollo de software.



# **Roles que puedes desempeñar**

- Site Reliability Engineer.

Se enfocan en que los sistemas de software operen de manera correcta y con el mayor grado de confiabilidad posible.



# **Roles que puedes desempeñar**

- Security Operations Engineer.

Encargados de mantener la seguridad de los sistemas a nivel de red y aplicaciones.





# Algunos otros roles

- Network Engineer
- Database Administrator
- Network Operation Center Engineer
- MLOps Engineer
- Cloud engineer

**¿Qué son  
los servidores?**

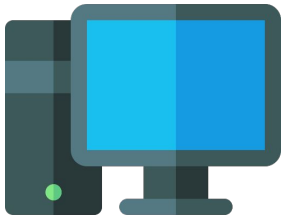


# ¿Qué es un servidor?

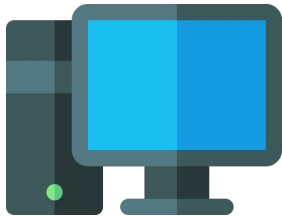
Un servidor es un **grupo de recursos tecnológicos** (hardware y software) que cumplen con uno o varios propósitos.

Estos usualmente reciben **peticiones de un cliente** y luego otorgan una **respuesta**.

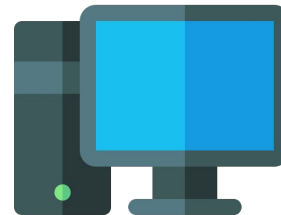
**Mainframe**



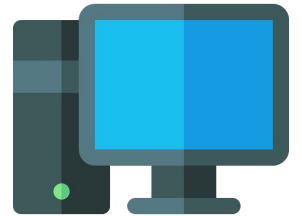
**Dumb  
terminal**



**Dumb  
terminal**



**Dumb  
terminal**



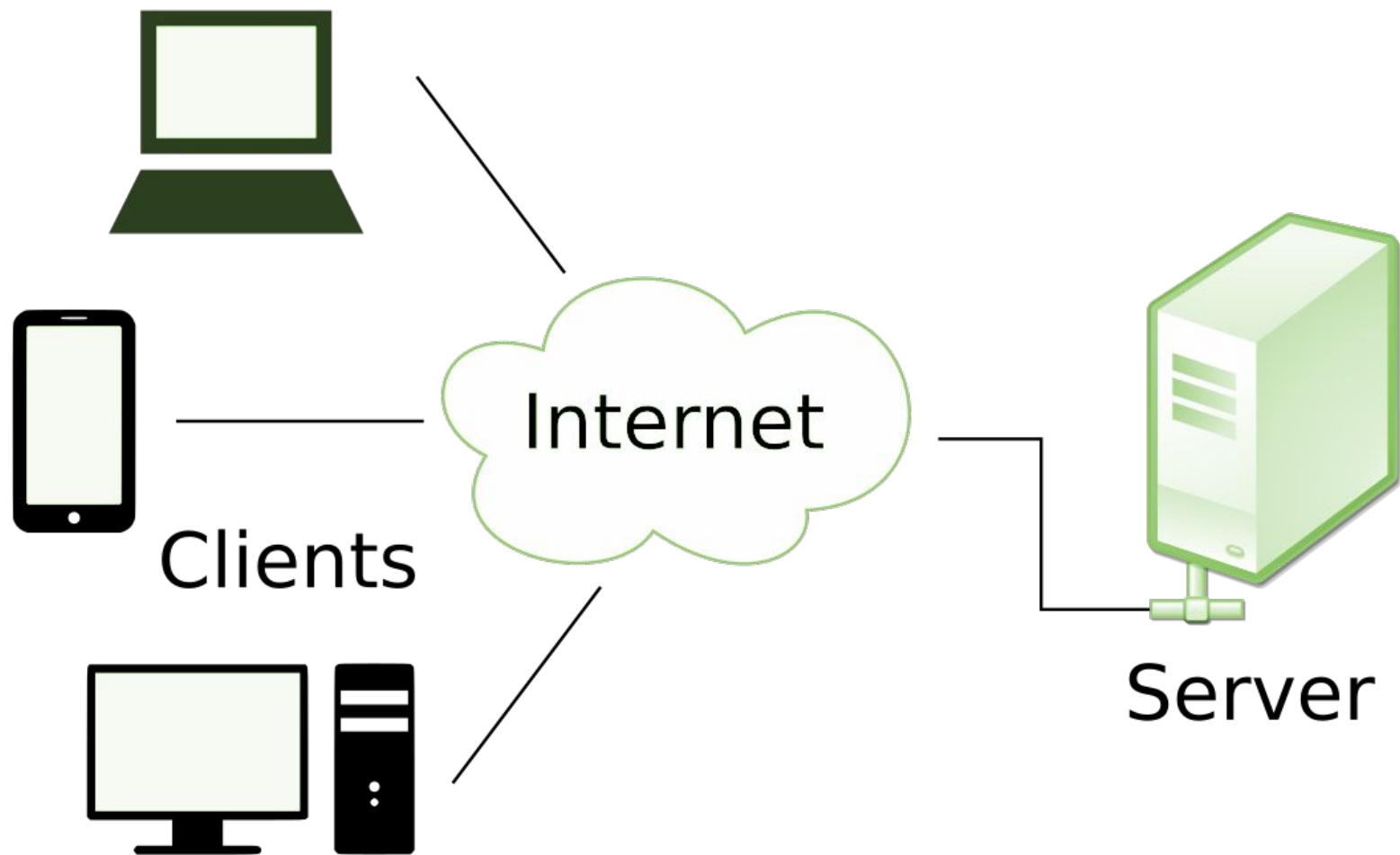
**Dumb  
terminal**





```
SLNIN .SYS 12P 20-Dec-85  VM .SYS 3P 13-Aug-86
XL .SYS 4P 20-Dec-85  LD .SYS 8P 23-Aug-86
SP .SYS 8P 13-Aug-86  ML .SYS 8P 13-Aug-86
RT11SJ.SYS 70P 13-Aug-86  MU .SYS 2P 13-Aug-86
KL .SYS 2P 13-Aug-86  TT .SYS 71P 21-Nov-84
SQ .SYS 8P 31-Aug-85  KLO296.SYS 24 20-Dec-85
BASIC.SAV 56 24-Aug-79  DIR .SAV 19 20-Dec-85
DATIME.SAV 4 20-Dec-85  DUP .SAV 47 20-Dec-85
DUMP .SAV 9 20-Dec-85  FORTRA.SAV 200 21-Nov-85
TSXKOD.SAV 70 27-Nov-82  LET .SAV 5 20-Dec-85
HAROLD.SAV 41 12-Jun-85  RETRO.OBJ 1515P 16-Nov-88
START.P36 2 21-Dec-91  TSXCL.TSX 22 07-Mar-82
ENIA .STW 15P 02-Feb-83  TSXV6.HSC 1P 04-Sep-85
STAND.LIN 12P 15-Aug-83  JKFLIP.SAV 30 08-Mar-86
EN1B .STW 19 11-Feb-83  RT11FD.SYS 83P 20-Dec-85
JKFLIP.FOR 3 06-Mar-86  TSXPF2.H39 1200P 27-Nov-82
ANNOT.SAV 30 18-Apr-83  BL .DIR 7 16-Jul-86
DNO .DIR 18 16-Jul-86  DEHDFC.OBJ 1 -BAD-
DIR .DIF 26 16-Jul-86  EVAN .ID 1
DEHDFC.OBJ 1 -BAD-
114 Files, 5949 Blocks
14433 Free blocks
.00
```

**Pero eso ya es  
viejísimo profe...**



# Caso de uso



**GEFORCE  
NOW™**





# Algunos tipos de servidores

- Web (frontend, backend)
- Bases de datos
- De pruebas
- Videojuegos
- Medios
- Email
- Impresión
- Archivos y recursos (SFTP, SMB)
- VoIP

**Y existen muchos más...**



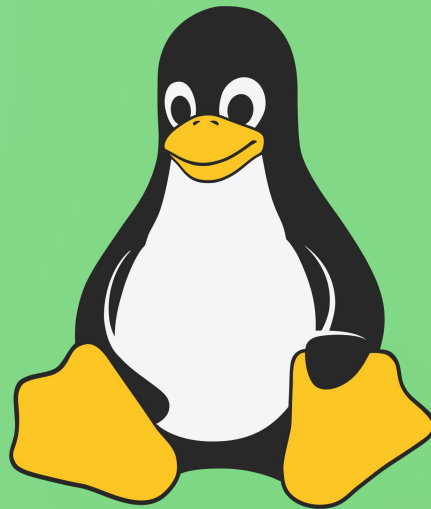


# Servidor de pruebas

Esto usa Uber para hacer pruebas de su aplicación en diferentes dispositivos móviles **reales**.



# Empecemos a aprender de Linux



**¿Qué es un sistema  
Linux/UNIX?**



**Linux es un kernel,  
que es solo una  
parte del sistema.**



# GNU/Linux

Es una familia de sistemas operativos que usan a Linux como Kernel y una gran colección de programas para conformar un sistema operativo completo.

# Distribuciones



**¿Cuál es la diferencia  
entre Linux y UNIX?**



# UNIX

UNIX es un sistema operativo  
que se desarrolló en el año de 1969  
por los laboratorios Bell de AT&T.

# Distribuciones



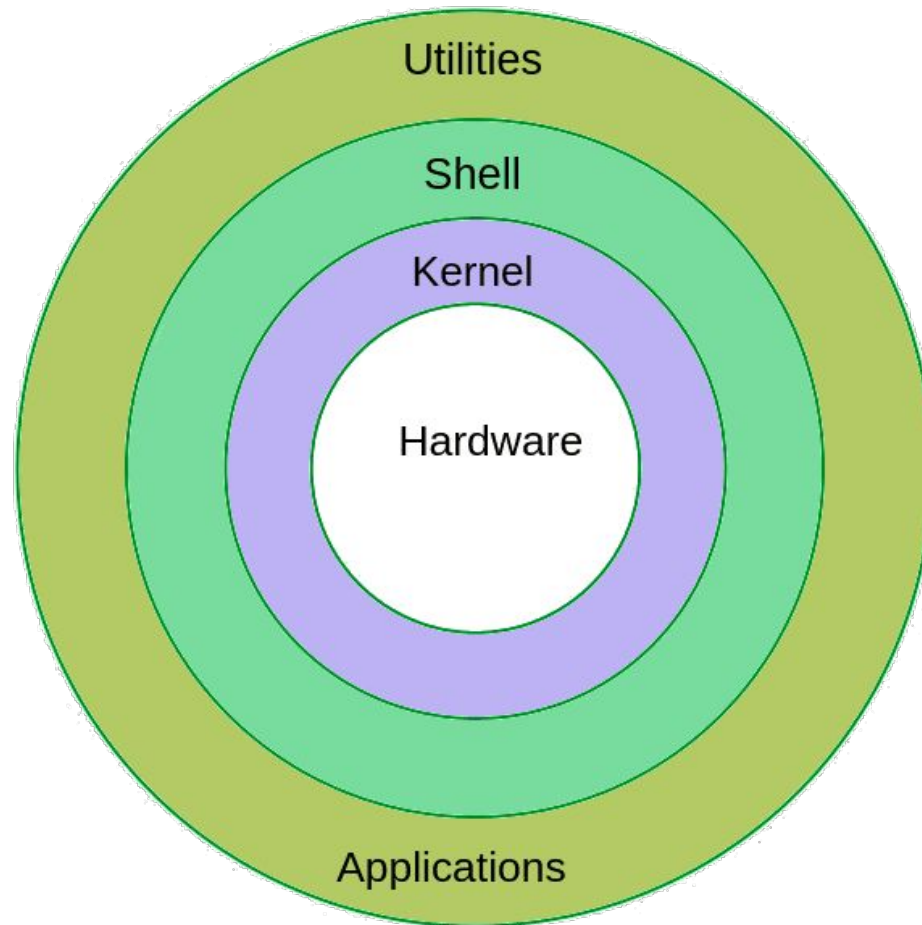


# **Arquitectura de un sistema UNIX/Linux**

**¿Cómo están  
formados?**



# Ambos tienen capas





# Hardware

Son todos nuestros dispositivos físicos conectados al sistema (discos, mouse, memoria, procesador, teclado, etc.).



# Kernel

Es la parte clave de todo sistema operativo Linux.

Es una pieza de software que nos permite controlar todo el hardware de nuestro servidor como el uso de CPU o memoria RAM.



# Shell

Es la interfaz que está entre el kernel y el usuario.

Es quien nos permite ejecutar comandos y pasarlos a un sistema de bajo nivel.

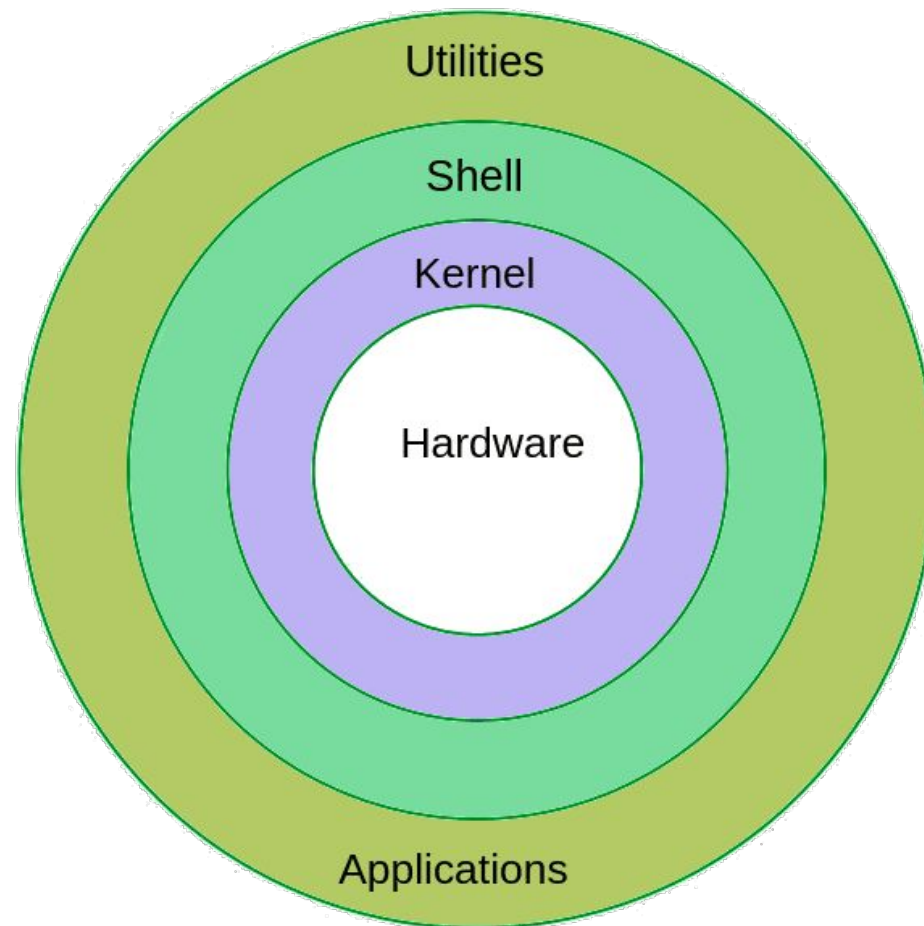




# Aplicaciones

Es donde el usuario directamente interactúa.  
Es la capa donde trabajan nuestros comandos y aplicaciones.

# Arquitectura de un sistema UNIX/Linux



# **Breve historia del software libre y el open-source**



# GNU (not Unix)



# Free Software Foundation

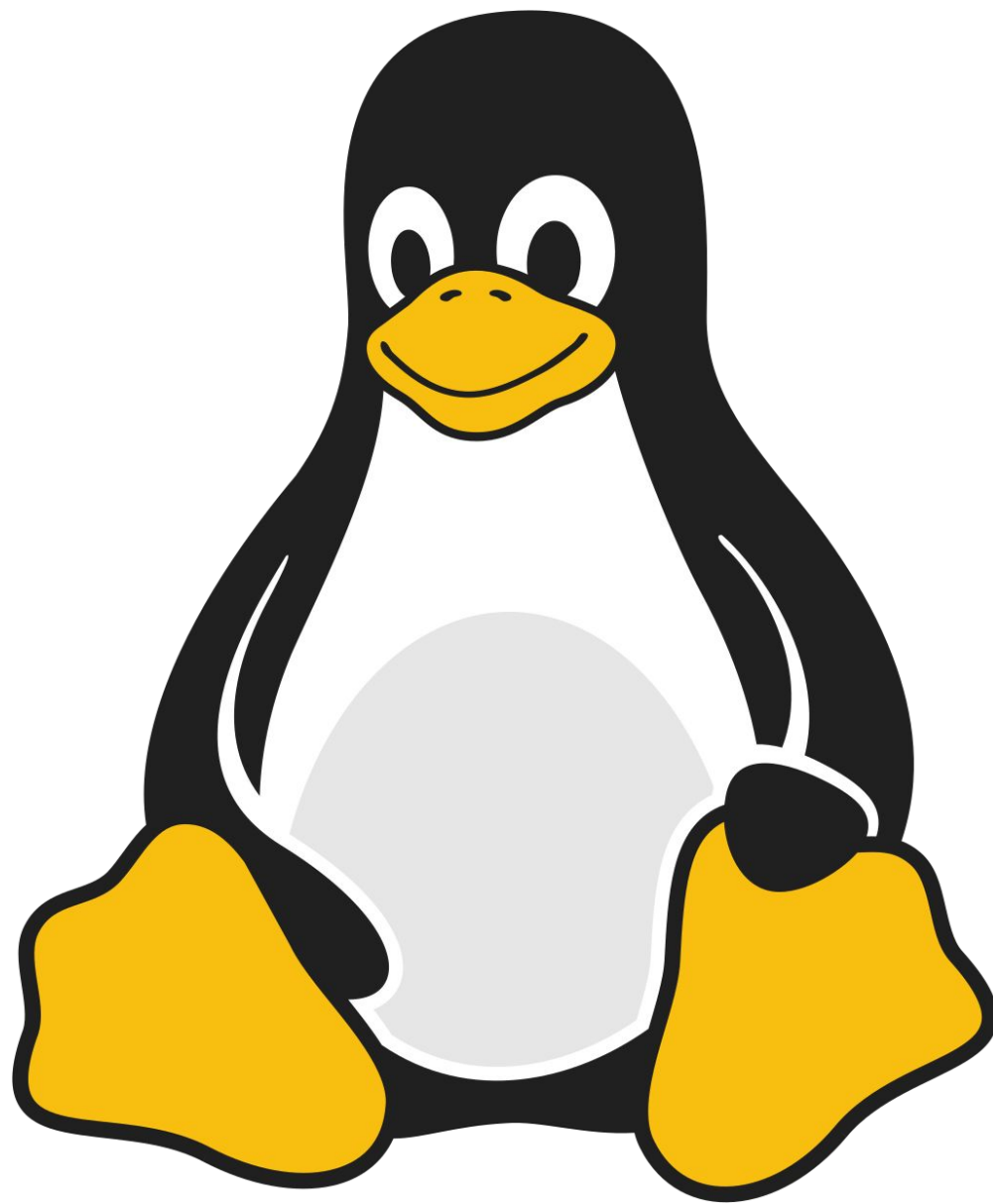




# Cuatro libertades del software libre

0. La libertad de ejecutar el programa.
1. La libertad de estudiar el funcionamiento del programa y modificarlo.
2. La libertad de redistribuir.
3. La libertad de distribuir copias de sus versiones modificadas a otras personas.





# Open Source



open source<sup>®</sup>  
initiative

# Ejemplo



**La diferencia radica  
en las libertades y  
filosofías de cada uno.**

# **Sistemas operativos y distribuciones**



# Distribución

Interpreta el Kernel de Linux, pueden variar en el formato, manejador de paquetes y popularidad.

Existen cientos de distribuciones independientes,  
pero las más populares son...





# LTS (long-term support)

Versiones de la distribución que prometen 5 años de soporte y mantenimiento.

Nota del autor: lo que recomendamos instalar.





# Non-free-repositories

Son paquetes con licencias no aprobadas por los lineamientos de Debian Free Software.



# Rolling Vs Fixed Release

Rolling release: las actualizaciones se publican al momento de desarrollarse. (Arch Linux, Gentoo-based).

Fixed release: las actualizaciones son probadas y posteriormente publicadas. (Ubuntu).

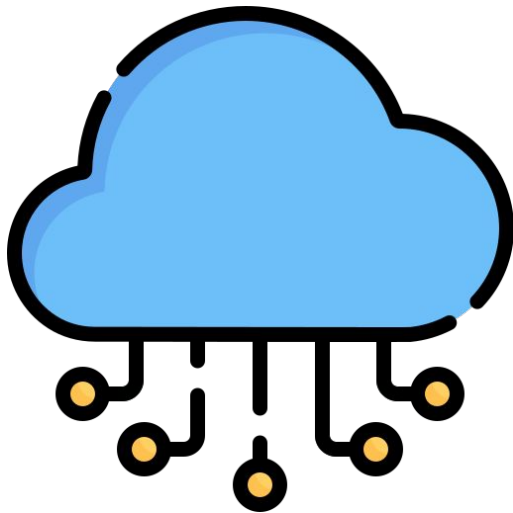
**¿Dónde viven  
nuestros servidores?**



# On premise

Todo el hardware y software del servidor es alojado y mantenido por la organización.


# Cloud



Pública



Privada



# Hybrid (Híbrida)

Es una combinación de servicios on premise y cloud.











# **Formas de montar un servidor**



# Instalación directa

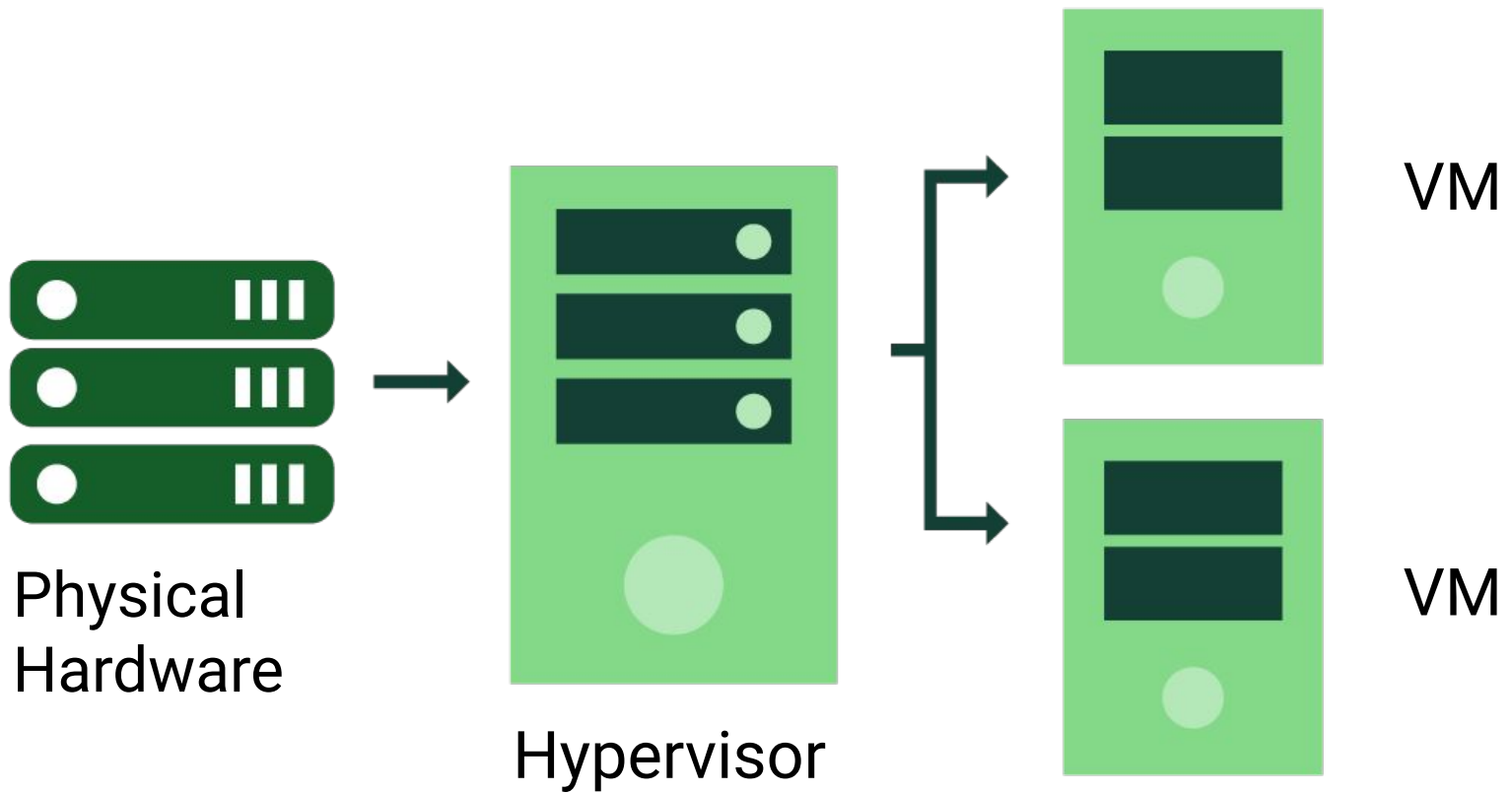
Se instala un sistema operativo para ocupar el 100% de los recursos dedicados (software y hardware) ya sea a un solo servicio o varios.



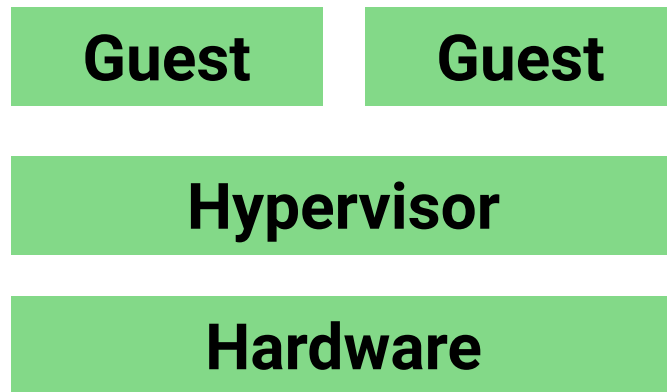
# Virtualización

Se instala un software que sirve como **host** conocido como (**hypervisor**) que administra los recursos para crear múltiples **guests**.

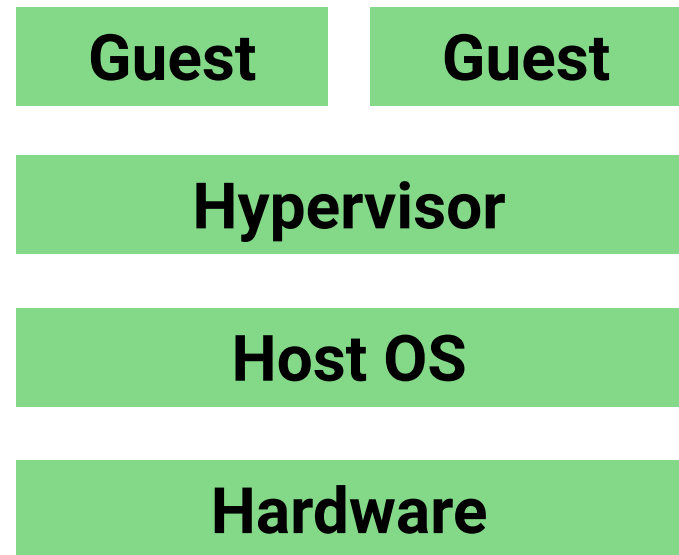
# Virtualization



# Hypervisors



Tipo 1  
(Bare-metal)



Tipo 2  
(Hosted)

# **Contenedores y máquinas virtuales**



# **Instalando Ubuntu Server**



# **Instalando RHEL**

# **Configuración básica para un servidor remoto**



# SSH

**Secure Shell Protocol:** Es un protocolo que nos permite acceder de manera remota a un servidor.

**¿Qué son los  
sistemas de  
archivos?**



# Sistema de archivos

Es un sistema que nos permite almacenar, mover, eliminar y buscar archivos.

Puede ser compatible o no entre diferentes sistemas operativos.



# Principales sistemas de archivos locales

- FAT
- FAT32
- NTFS
- EXT4
- XFS
- Btrfs
- ZFS

# **Particiones de un servidor Linux**

# Árbol de directorios



# **Diferentes tipos de archivos**

# Tipos de archivo

Tipo de archivo	Símbolo	Crear	Remover
Archivo regular	-	editores, cp,...	rm
Directorio	d	mkdir	rmdir, rm -r
Archivo de caracteres	c	mknod	rm
Archivo de bloque	b	mknod	rm
Socket de dominio local	s	socket system call	rm
Named Pipe	p	mknod	rm
Link simbólico	l	ln -s	rm



# Archivos regulares

Consiste en una serie de bytes, el sistema de archivos no le impone una estructura.

Archivos de texto, archivos de datos, programas ejecutables y librerías.



# Directorios

Contienen referencias a otros archivos.

/. referencia al directorio mismo

/.. referencia al directorio derivado



# Hard links

Estos links son una copia de un archivo original, hacen referencia a un punto de la memoria en la que se almacena el archivo.



# Links simbólicos

También se conocen como soft links.  
Apuntan al archivo original, por una referencia por nombre.

Si eliminamos el archivo original, el link queda inservible, a diferencia de los hard links.



# Archivos de dispositivos

De caracteres/bloque.

Estos archivos permiten que los programas se comuniquen con los periféricos y el hardware.

No son drivers, pero sí definen la comunicación con ellos y el manejo del dispositivo.



# Sockets de dominio local

Son conexiones entre procesos que intermedian la comunicación, sobre todo lo que involucra la red.

Solo se puede acceder a ellos desde el localhost.





# Named Pipes

Estos archivos comunican dos procesos que corren al mismo tiempo.

Se conocen como FIFO files (first in/first out), que pueden entenderse como una pila de procesos.

# **Conociendo los repositorios y paquetes**



# Paquetes

Incluyen todos los archivos necesarios para ejecutar el software, hacen el proceso de instalación lo más sencillo posible, porque incluye los archivos binarios, de configuración y dependencias.



# Repositorio

Almacena los paquetes para que el usuario pueda descargarlos e instalar el software.

Pertenecen a los distribuidores de Linux, aquí se liberan las actualizaciones de los paquetes.



# .deb

Formato de instalación de paquetes de Debian y Ubuntu.

**dpkg**: herramienta que instala, desinstala y consulta.



# .rpm

Formato de instalación de paquetes de Red Hat, CentOS, SUSE, Amazon Linux.

**rpm:** herramienta que instala, desinstala y consulta.



# rpm y dpkg

## Comandos all-in-one

- install
- remove
- l (list)
- i (install)
- q (query, acompaña con una bandera)
- U (upgrade)
- e (erase)

**¿Qué es un manejador  
de paquetes?**



# Manejadores de paquetes



APT



yum/dnf

# **Aprende a usar el manejador de paquetes**

# **Manejo de paquetes a profundidad**

# **¿Cómo instalar software?**

**¿Qué es un proceso  
en Linux?**



# Proceso

Un proceso representa una referencia a un programa ejecutándose dentro de nuestro sistema, el cual puede ser controlado y monitoreado.

Este hace uso de recursos de memoria, de I/O, de procesamiento, etc.



# Demonio

Son procesos que se ejecutan en segundo plano, ejecutándose de forma persistente o reiniciándose de forma automática bajo ciertas condiciones.

Usualmente, se inician en el arranque del sistema.



# Señales

Son procesos de bajo nivel que sirven como peticiones en el ámbito de kernel para interrumpir otros procesos.



# **Escaneo de procesos**

# **Manejo de procesos**



# Estados de un proceso

- Running or Runnable (R)
- Uninterruptible Sleep (D)
- Interruptible Sleep (S)
- Stopped (T)
- Zombie (Z)

# **Creación y manejo de demonios**

# **Automatización de procesos**

**Esto es solo  
el comienzo**