

SISTEMAS OPERATIVOS II



BIENVENIDA

Bienvenido(a) a la asignatura *Sistemas Operativos II*, con la cual continuarás con el aprendizaje del sistema operativo Linux que comenzaste en la materia *Sistemas Operativos I*.

En la presente asignatura aprenderás a implementar la seguridad en la red y servidores de Linux. Además, conocerás cómo distribuir el sistema y el proceso de multitareas, así como la resolución de problemas de red y el desempeño y almacenamiento de AWS, y aplicaciones con DevOps.



LIBROS RECOMENDADOS

Alapati, S. R. (2016) Modern Linux Administration: how to become a cutting-Edge Linux administrator. O'Reilly.

Matotek, D., Turnbull, J., & Lieverdink, P. (2017) Pro Linux System Administration: learn to build systems for your business using free and open source software. Apress.

UNIDAD 1

SEGURIDAD





1.1

TEMARIO

1.2

SEGURIDAD
DE LA RED

SEGURIDAD
DEL SERVIDOR



INTRODUCCIÓN

La asignatura *Sistemas Operativos II* tiene como objetivo continuar con el aprendizaje del sistema operativo de Linux. En esta ocasión, verás temas de seguridad, gestión de carga de trabajo, resolución de problemas y gestión de entornos externos.

En esta primera unidad aprenderás a identificar las amenazas que pudieran presentarse en la red, así como la manera de dar frente a estas. Además, aprenderás a mantener un servidor de manera segura por medio de varias recomendaciones.

COMPETENCIAS A DESARROLLAR



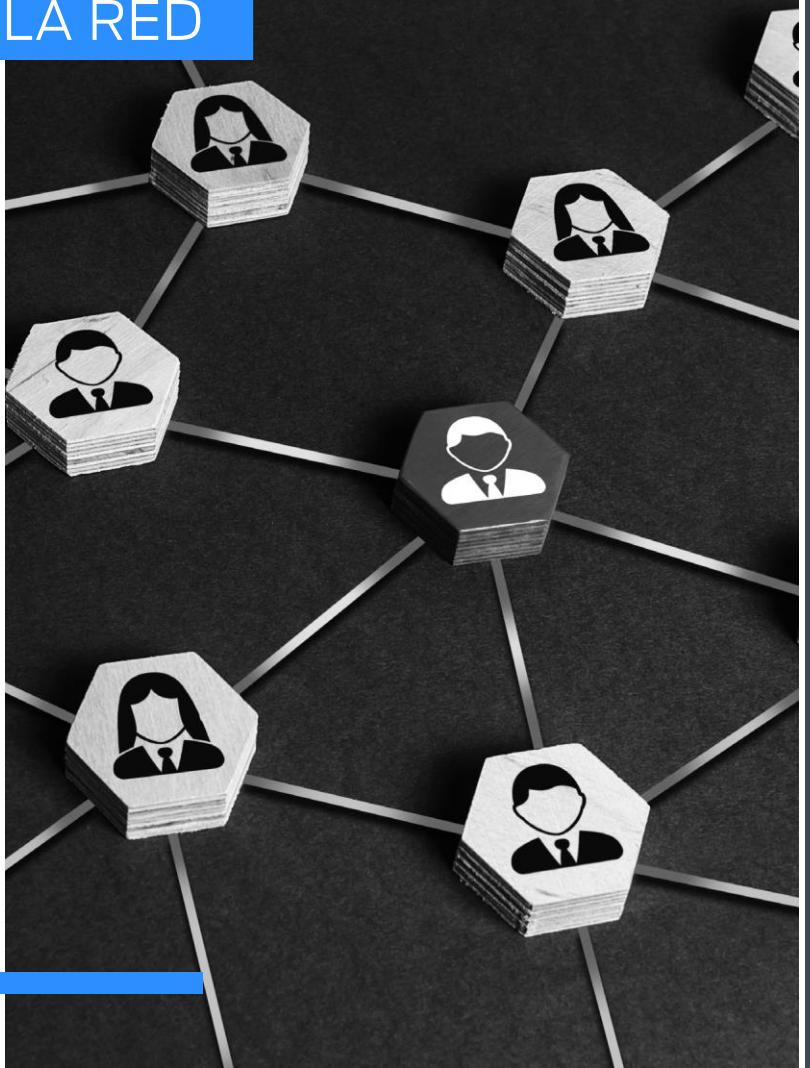
- El alumno será capaz de conocer las diferentes amenazas para el sistema operativo Linux y cómo prevenirlas.

- El alumno será capaz de conocer las medidas de seguridad de un servidor Linux.

SEGURIDAD DE LA RED

Las **amenazas de programas** suelen utilizar una falla en los mecanismos de protección de un sistema para atacar programas. Por el contrario, las **amenazas del sistema y de la red** implican el abuso de servicios y conexiones de red. Además, estas amenazas crean una situación en la que se utilizan indebidamente los recursos del sistema operativo y los archivos del usuario.

A veces, se utiliza un ataque de sistema y de red para lanzar un ataque de programa, y viceversa.





MISTAKE

Cuanto más abierto es un sistema operativo, cuantos más servicios ha habilitado y más funciones permite, más probable es que haya un error disponible para explotar.

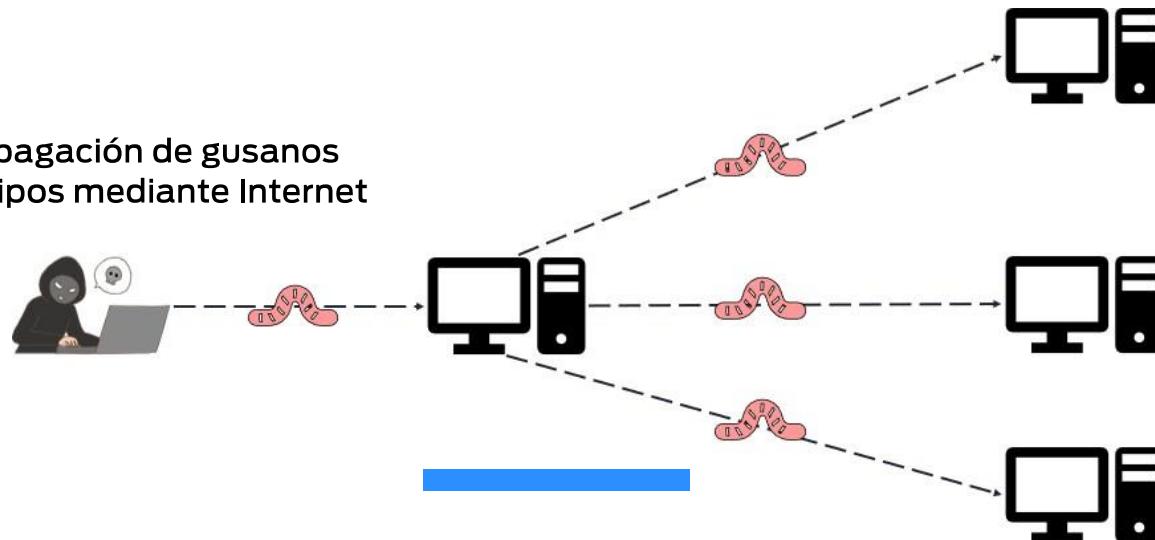
Por ejemplo: Solaris 10 pasó de ser un modelo en el que muchos servicios estaban habilitados de manera predeterminada cuando se instala el sistema a ser un modelo en el que casi todos los servicios están deshabilitados en la instalación. Por ello, hay que habilitarlos específicamente por el sistema de administradores.

A continuación, vamos a analizar algunos **ejemplos de amenazas de la red** y del sistema. Es importante tener en cuenta que los ataques de enmascaramiento y reproducción también se lanzan comúnmente a través de redes entre sistemas. De hecho, estos ataques son más efectivos y más difíciles de contrarrestar cuando están involucrados varios sistemas.



Gusanos (worms). Representan un proceso que usa el mecanismo de generación para duplicarse. Así, un gusano genera copias de sí mismo, utilizando los recursos del sistema y quizás bloqueando todos los demás procesos. En las redes informáticas, los gusanos son particularmente potentes, ya que pueden reproducirse entre los sistemas y, por lo tanto, apagar una red completa.

Tal evento ocurrió en 1988 en los sistemas UNIX en Internet, causando la pérdida de tiempo del sistema y del administrador, con pérdidas valuadas en un millón de dólares.



Método de propagación de gusanos por Morris.

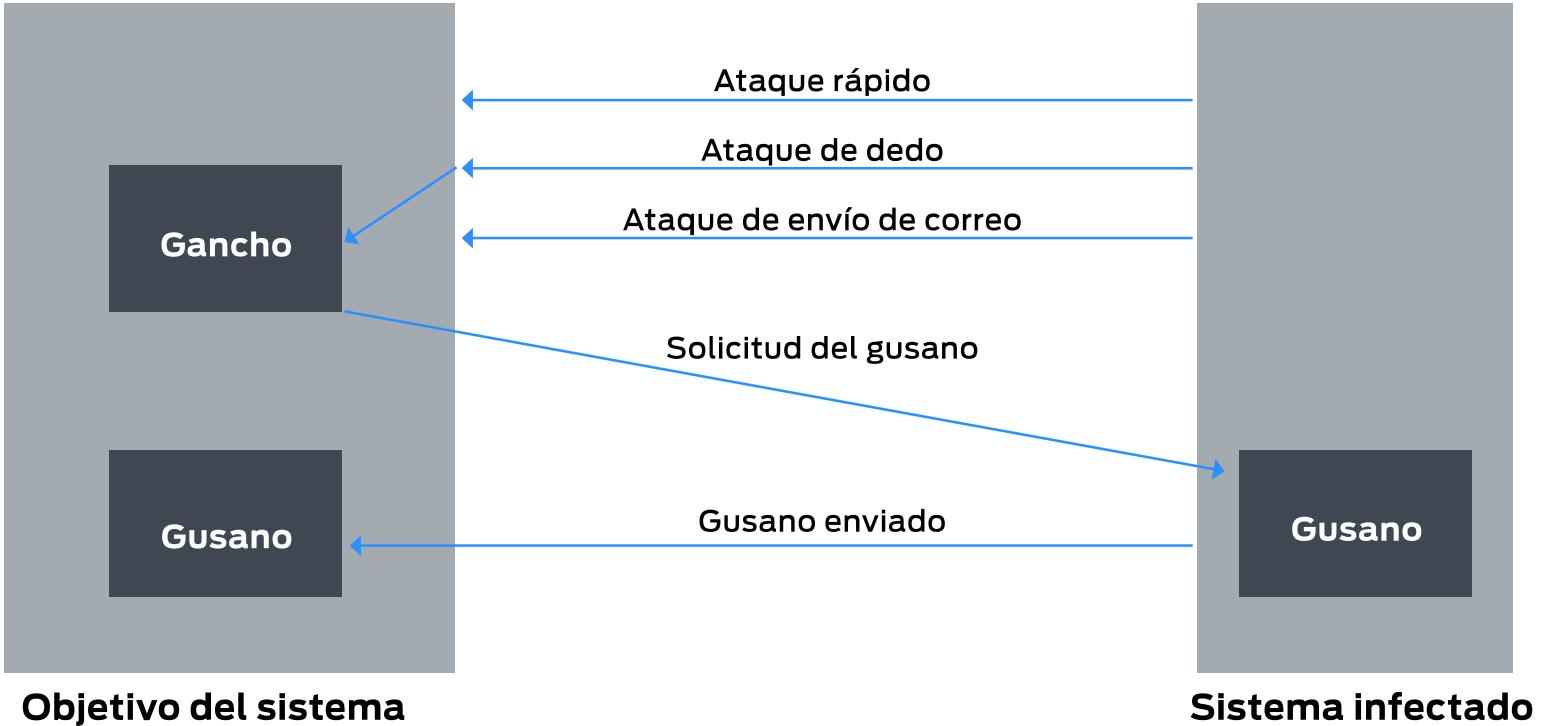
Antes que nada, es importante destacar que Robert Tappan Morris fue el responsable de enviar los gusanos en 1988 en los sistemas UNIX a través de Internet. Él especifica el **método que utilizó para tal propagación:**

“El gusano se componía de dos programas: un programa de gancho de agarre, también llamado *bootstrap* o vector, y el programa principal. El primero (llamado *l1.c*) constaba de 99 líneas de código C compiladas y ejecutadas en cada máquina a la que accedía (...”).

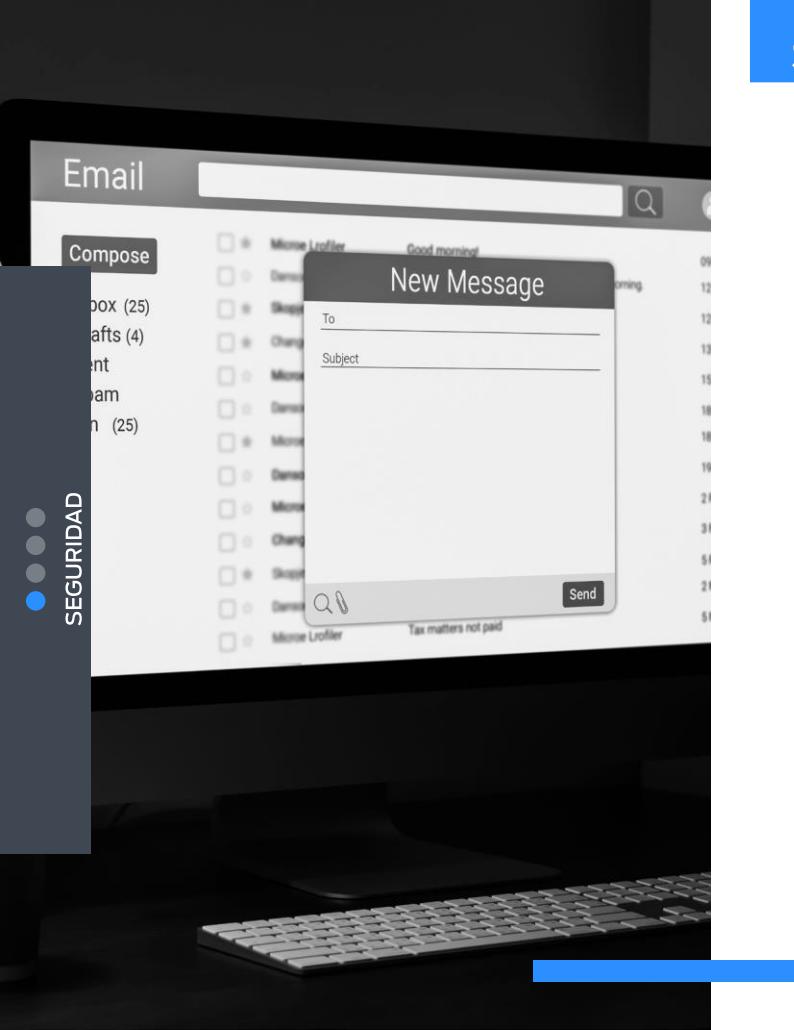




“ (...) Una vez establecido en el sistema bajo ataque, el garfio se conectó a la máquina donde se originó y cargó una copia del gusano principal en el sistema enganchado. Después, este procedió a buscar otras máquinas a las que el sistema recién infectado pudiera conectarse fácilmente. En estas acciones, se aprovecha la utilidad de red rsh de UNIX para facilitar la ejecución remota de tareas. Al configurar archivos especiales que enumeran los pares de nombres de inicio de sesión del host, los usuarios pueden omitir ingresar una contraseña cada vez que se accede a una cuenta remota en la lista”.



Funcionamiento del gusano de Internet de Morris



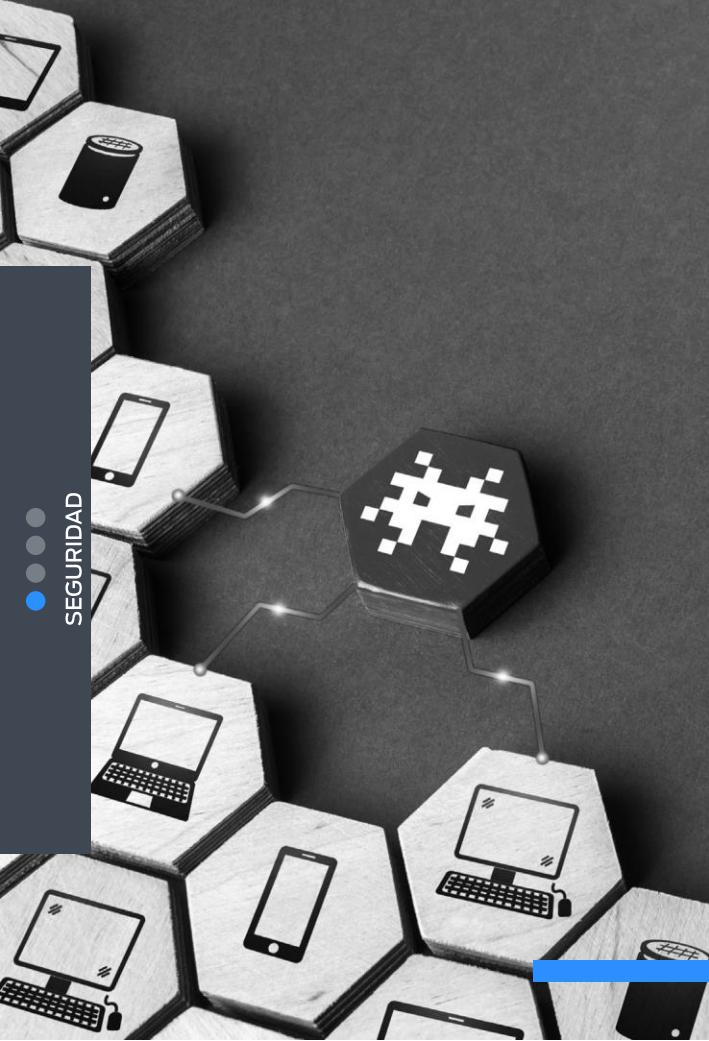
Escaneo de puertos. No es propiamente un ataque, sino un medio para que un *cracker* detecte las vulnerabilidades de un sistema. La exploración de puertos normalmente está automatizada e involucra una herramienta que intenta crear una conexión TCP/IP a un puerto específico o a un rango de puertos.

Por ejemplo, supongamos que existe una vulnerabilidad conocida en el envío de correos. En consecuencia, un *cracker* podría lanzar un escáner de puertos para intentar conectarse, por ejemplo, al puerto 25 de un sistema en particular.

Si la conexión fue exitosa, el *cracker* podría intentar comunicarse con el servicio de contestador para determinar si el servicio enviaba correo y, de ser así, saber si era la versión con el error.

Por otro lado, imagina un puerto en el que se codifique cada error de cada servicio de cada SO. Así, un *cracker* podría intentar conectarse a todos los puertos de uno o más sistemas. Por cada servicio que respondió, podría intentar usar cada error conocido.





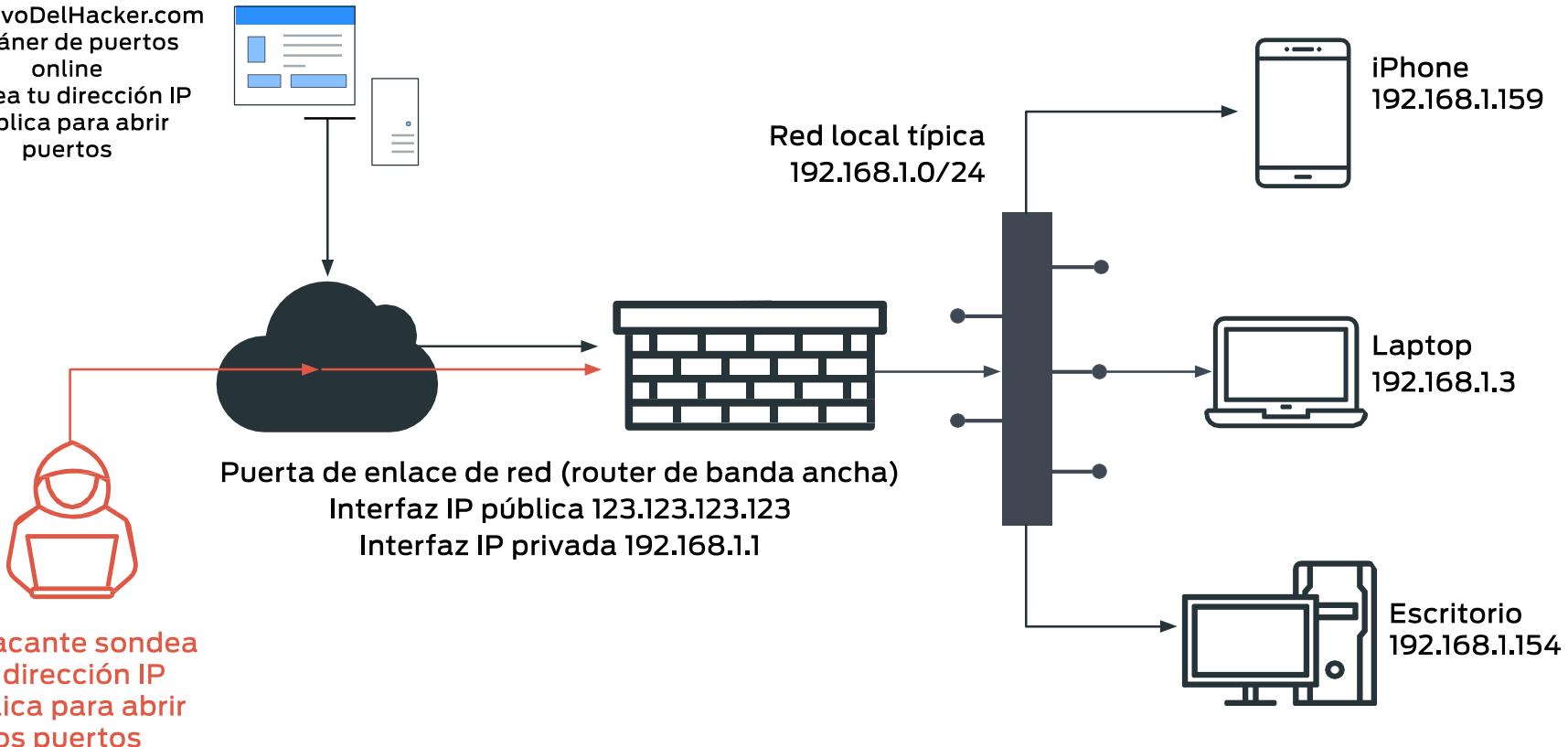
Con frecuencia, los errores son **desbordamientos de búfer**, lo que permite la creación de un *Shell* de comandos privilegiado en el sistema.

A partir de ahí, por supuesto, el *cracker* podría instalar caballos de Troya o virus troyanos, programas de puerta trasera, etcétera.

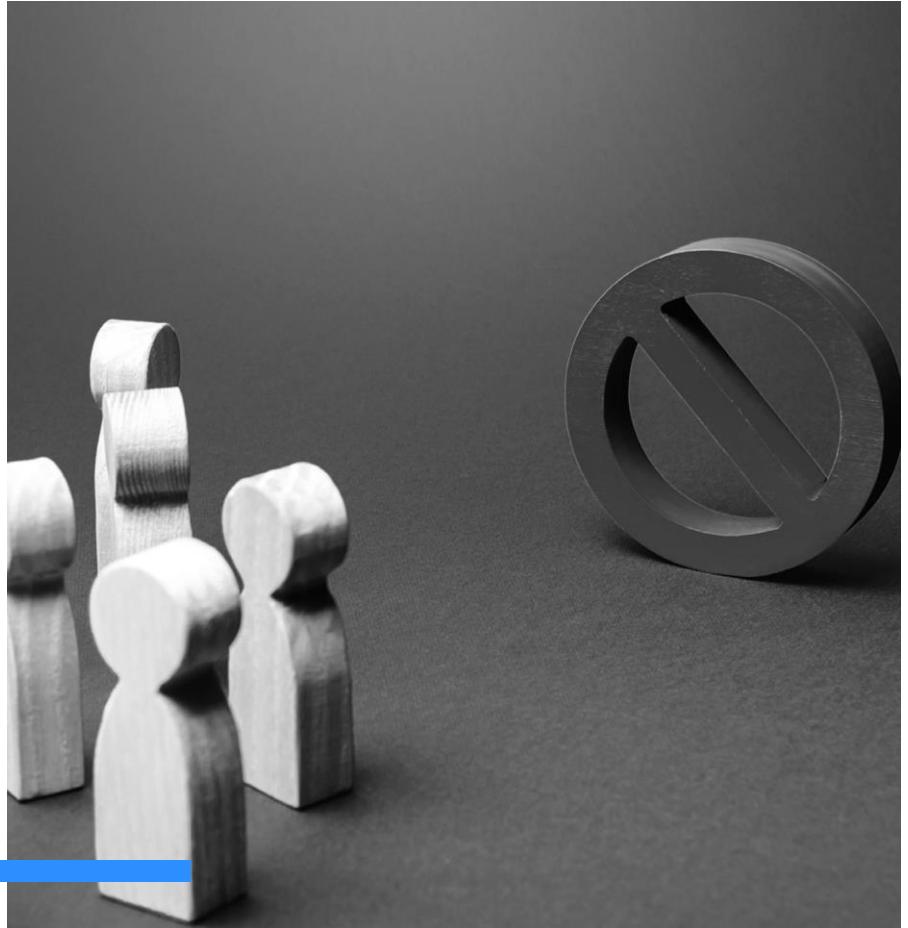
SEGURIDAD DE LA RED

SEGURIDAD

ObjetivoDelHacker.com
Escáner de puertos online
Testea tu dirección IP pública para abrir puertos



Denegación de servicio. No tienen como objetivo obtener información o robar recursos, sino interrumpir el uso legítimo de un sistema o instalación. La mayoría de estos ataques involucran sistemas que el atacante no ha penetrado. Lanzar un ataque que impida el uso legítimo suele ser más fácil que irrumpir en una máquina o instalación.

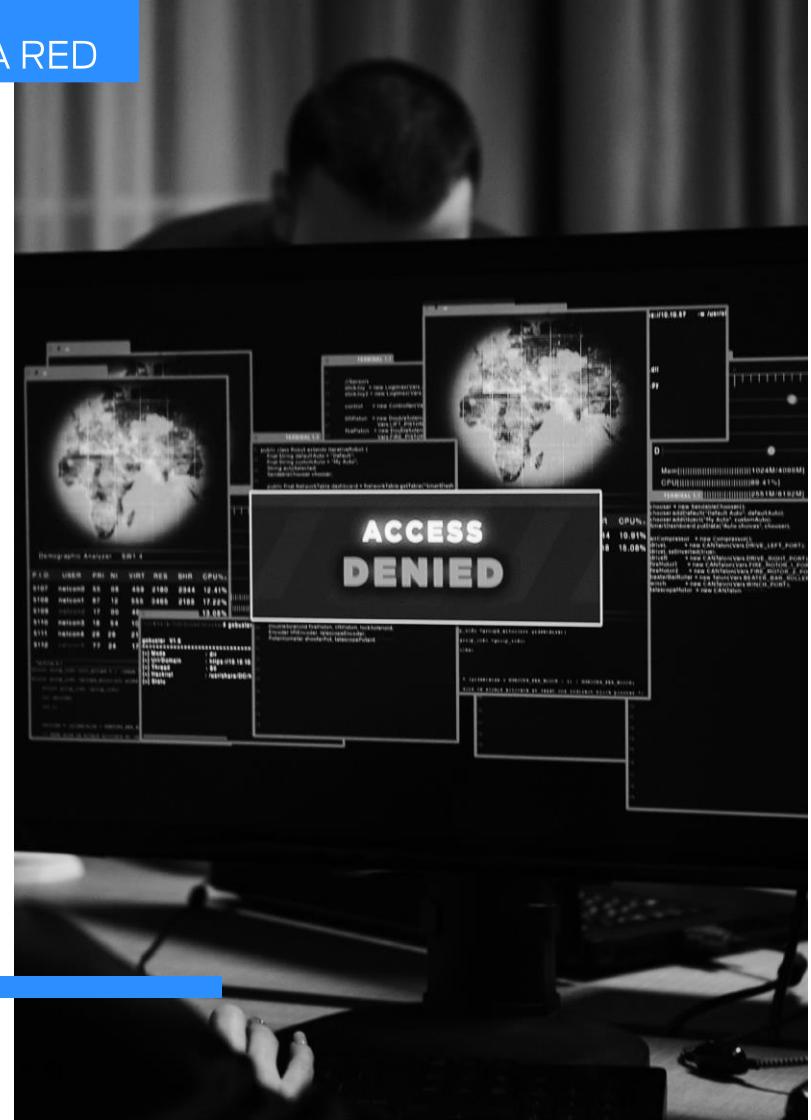




Los ataques de denegación de servicio generalmente se basan en la red. **Se dividen en 2 categorías:**

- **Primera categoría.** Utilizan tantos recursos de las instalaciones que, en esencia, no se puede realizar ningún trabajo útil.
- **Segunda categoría.** Involucran la interrupción de la red de la instalación.

Los ataques de denegación de servicio resultan del abuso de algunas de las funciones fundamentales de TCP/IP. Por ejemplo, si el atacante envía la parte del protocolo que dice “Quiero iniciar una conexión TCP”, pero nunca sigue con el estándar “La conexión ahora está completa”, el resultado puede resultar en sesiones TCP parcialmente iniciadas.





SEGURIDAD DE LA RED

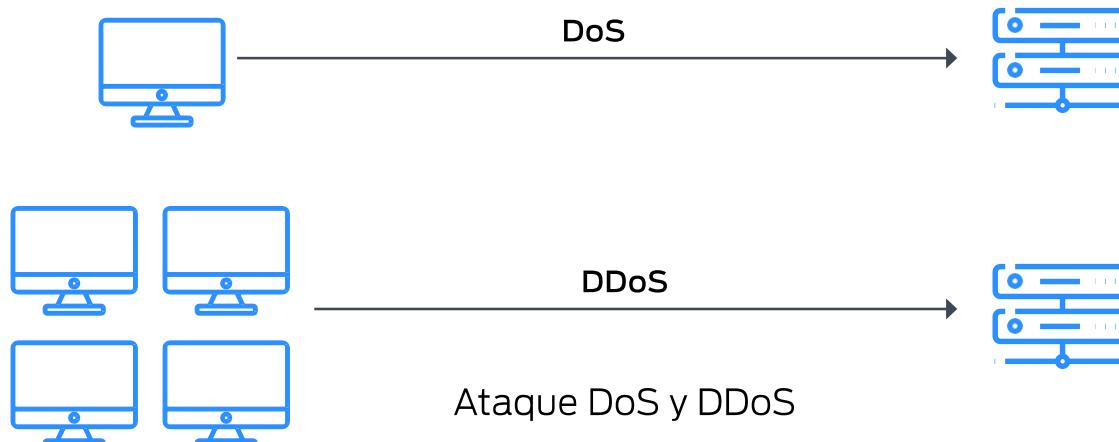
Si se inician suficientes de estas sesiones, pueden consumir todos los recursos de red del sistema, deshabilitando cualquier otra conexión TCP legítima.

Además, los ataques de denegación de servicio pueden durar horas o días.

Los ataques suelen detenerse a nivel de red hasta que los sistemas operativos puedan actualizarse para reducir su vulnerabilidad.

Generalmente, **es imposible prevenir los ataques de denegación de servicio**, dado que utilizan los mismos mecanismos que el funcionamiento normal. Aún más difíciles de prevenir y resolver son los **ataques de denegación de servicio distribuido (DDoS)**, ya que se lanzan desde múltiples sitios a la vez.

Los ataques DDoS se han vuelto más comunes y, a veces, se asocian con **intentos de chantaje**. En este sentido, un sitio es atacado y los atacantes ofrecen detener el ataque a cambio de dinero.





Defensas

Ante estas amenazas, ¿cómo podemos hacer que los sistemas sean seguros? En realidad, esto es posible, y veremos algunas de las formas en que se pueden diseñar e implementar sistemas para aumentar su seguridad.

Uno de los conceptos más importantes es la **defensa en profundidad**. Básicamente, nos habla de tener varias capas de seguridad para que, si se viola una de ellas, haya otras que superar.

Firewalls. Es como una adaptación moderna del recurso de seguridad medieval: cavar un foso profundo alrededor del castillo. Este diseño obligaba a todos los que entraban o salían a pasar por un solo puente levadizo, donde podían ser inspeccionados por la policía I/O.

Con las redes, es posible el mismo truco: una empresa puede tener muchas LAN conectadas, pero todo el tráfico hacia o desde la empresa se fuerza a través de un puente levadizo electrónico: el *firewall*.



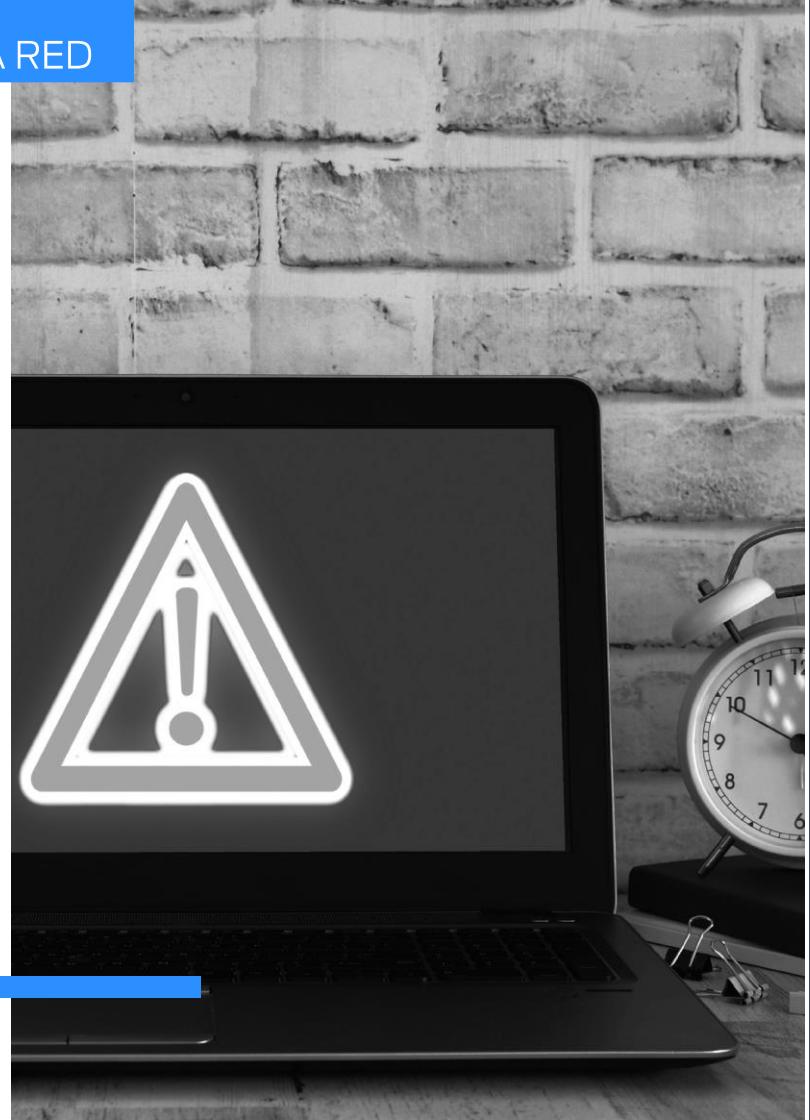


SEGURIDAD DE LA RED

Antivirus. También combaten gusanos y *spyware*. Los virus intentan esconderse y los usuarios intentan encontrarlos, lo que lleva a un juego parecido al del gato y el ratón. En este sentido, los virus son como los *rootkits*, excepto que la mayoría de los creadores de virus enfatizan su rápida propagación en lugar de jugar al escondite en la maleza como lo hacen los *rootkits*.

Algunos **ejemplos de antivirus** son los siguientes:

- Escáneres de virus
- Comprobadores de integridad
- Comprobadores de comportamiento
- Prevención de virus



SEGURIDAD DEL SERVIDOR

En Linux, el aseguramiento de los servidores es una tarea sumamente importante. Con ello, podemos garantizar la protección de los datos, propiedad intelectual y tiempo de los *hackers*.

Bajo esta perspectiva, el **administrador del sistema** es la persona responsable de la seguridad de la caja de Linux.



VIDEO

Te invitamos a ver el siguiente video:





Cifrado de la Comunicación de Datos para el Servidor

Todos los datos transmitidos a través de una red están abiertos a monitoreo.

Por lo anterior, para cifrar los datos que son transmitidos continuamente, es viable contar con una contraseña; o bien, utilizar claves/certificados.

Recomendaciones para el cifrado de la comunicación de datos para el servidor:

- **GnuPG** te permite encriptar y firmar sus datos y comunicaciones.
- Utilice **scp**, **ssh**, **rsync** o **sftp** para la transferencia de archivos.
- Utiliza **OpenVPN**, ya que es una VPN SSL efectiva y rentable.
- **Lighttpd SSL** (Secure Server Layer) para la configuración e instalación de HTTP
- Configuración e instalación de **Apache SSL** (Secure Server Layer) Https (mod_ssl)



IMPORTANTE. Evita el uso de los servicios FTP, Telnet y Rlogin / Rsh en Linux.

En casi cada una de las configuraciones de red, cualquiera que se encuentre conectado a la misma red puede capturar usuarios, contraseñas, comandos [FTP/telnet/rsh](#) y archivos que son transferidos, por medio de un **rastreador de paquetes**. La solución común a este problema es utilizar **OpenSSH, SFTP o FTPS**, que añadieron SSL o TLS a FTP:

```
# yum erase xinetd ypserv tftp-server telnet-server rsh-server
```

No obstante, si se está utilizando un servidor basado en Debian/Ubuntu Linux, prueba el comando [apt-get](#) / comando [apt](#) para eliminar los servicios inseguros:

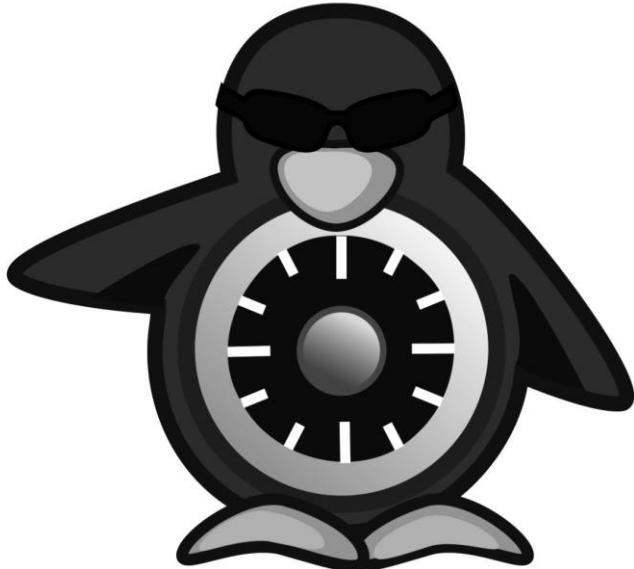
```
$ sudo apt-get -purge remove xinetd nis yp-to-los tftpd atftpd tftpd-hpa telnetd  
rsh-server rsh-redone-server
```

Mantener el Kernel y software de Linux actualizados:

Linux ofrece cada una de las **herramientas importantes** para conservar un sistema actualizado, y posibilita actualizaciones sencillas entre variantes.

Por ello, toda actualización de seguridad debe revisarse y aplicarse lo antes posible. Utiliza el administrador de paquetes RPM como [yum](#) y/o [apt-get](#) y/o [dpkg](#) para utilizar cada una de las actualizaciones de estabilidad.





SELinux:

Se recomienda su uso, ya que otorga un control de ingreso forzoso (MAC) flexible. De acuerdo con el control de ingreso discrecional (DAC) de Linux estándar, una aplicación o proceso que se realiza como cliente (UID o SUID) tiene los papeles del cliente para objetos como archivos, *sockets* y otros procesos.

Configurar el envejecimiento de la contraseña para usuarios de Linux para una mejor estabilidad:

El comando **chage** se encarga de cambiar el número de días entre los cambios de contraseña y la fecha del último cambio de contraseña. El documento **/etc/login.defs**, por su parte, define la configuración específica del lugar para el grupo de contraseñas escondidas, incluida la configuración de antigüedad de la contraseña. El siguiente comando sirve para **deshabilitar la caducidad de la contraseña**:

```
# chage -M 99999 userName
```

Para obtener **información sobre la caducidad de la contraseña**, ingresa:

```
# chage -1 userName
```



Restringir el uso de contraseñas anteriores en Linux:

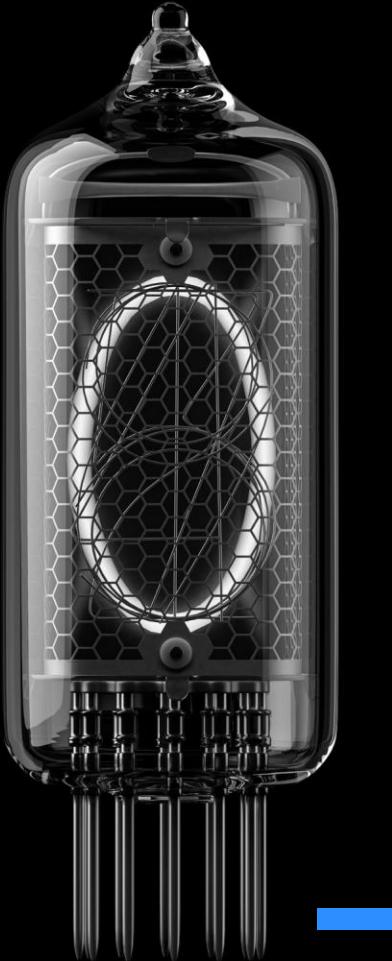
Además, puedes evitar que todos los usuarios usen o reutilicen las mismas contraseñas antiguas en Linux. El parámetro del módulo [pam_unix](#) [remember](#) se puede usar para configurar el número de contraseñas anteriores, de manera que no se pueden volver a utilizar.

Bloqueo de cuentas de cliente luego de fallas de inicio de sesión:

En Linux se puede utilizar el comando `faillog` para demostrar o establecer los límites de falla de inicio de sesión, o implantar parámetros de fracaso de inicio de sesión. Este comando se encarga de formatear el contenido del registro de fallas a partir de `/ var / log / faillog database / log file`.

Para **desbloquear una cuenta después de fallas de inicio de sesión**, ejecuta lo siguiente:

```
faillog -r -u userName
```



Asegúrate de que ninguna cuenta NO root tenga el UID establecido en 0:

Solo la cuenta del *root* tiene **UID** (código identificador utilizado en informática: *users ID*) “0” con todos los permisos que hay para poder acceder al sistema.

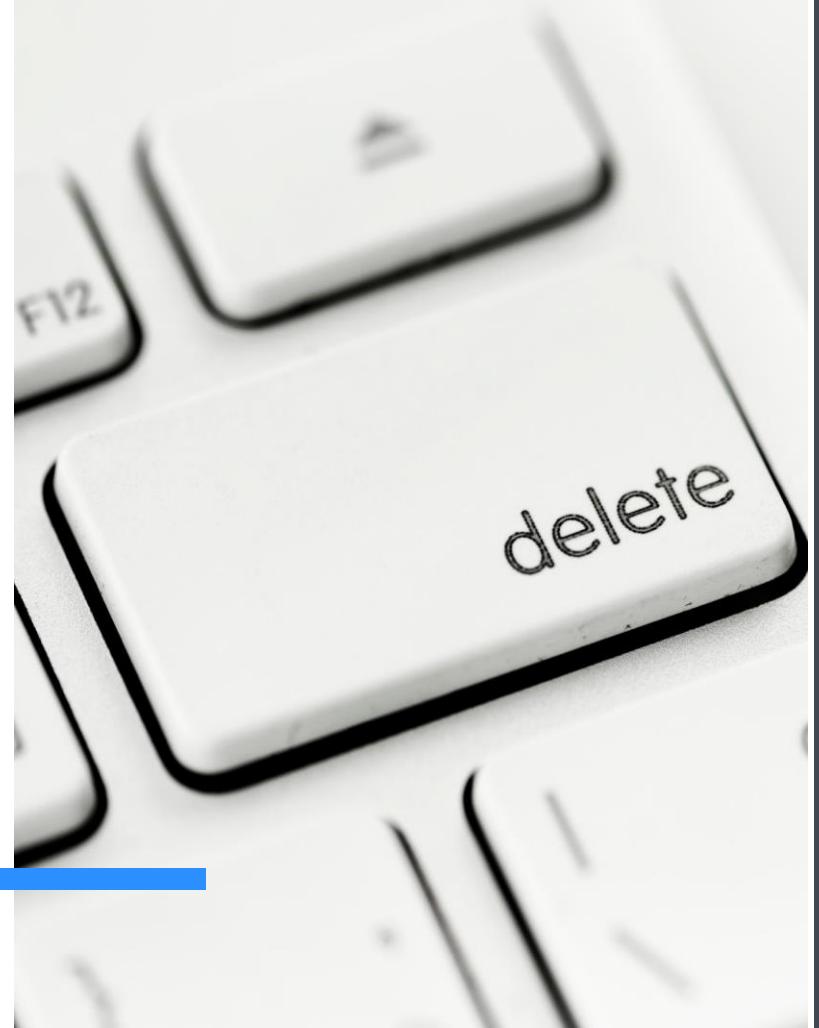
Escribe el siguiente comando para mostrar todas las cuentas con UID establecidas en “0”:

```
# awk -F: '($3 == "0") {print}' /etc/passwd
```

Desactiva los servicios de Linux no deseados:

Es importante deshabilitar todos los servicios y *demonios (daemons)* que sean innecesarios; es decir: los servicios que se ejecutan en segundo plano.

Para ello, es esencial que elimines todos los servicios no deseados cuando se inicia el sistema.



Escribe el siguiente comando para **enumerar todos los servicios que se inician en el momento del arranque en el nivel de ejecución # 3**:

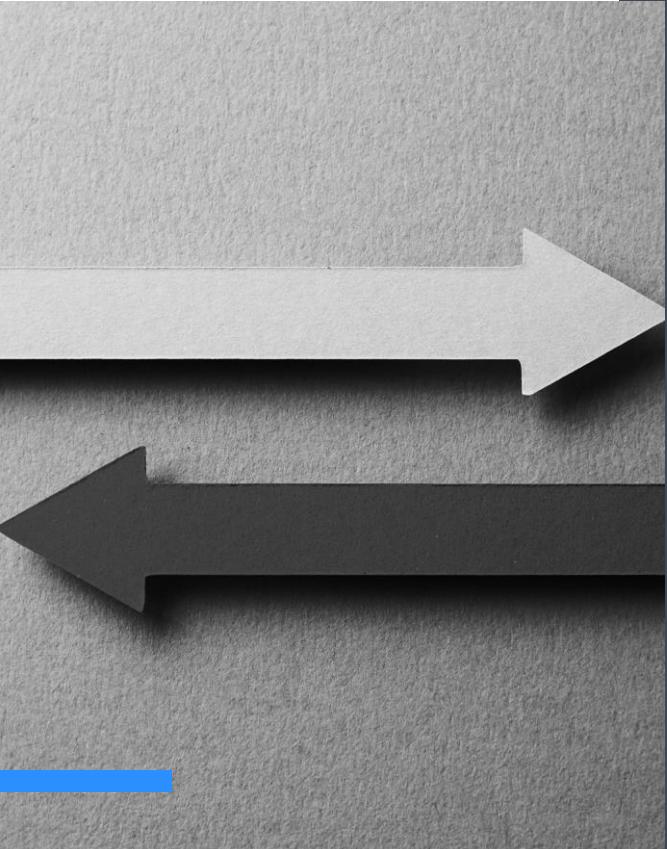
```
# chkconfig -list | grep 3:on'
```

Y para **deshabilitar el servicio** escribe lo siguiente:

```
# service serviceName stop
# chkconfig serviceName off
```

Configura el firewall basado en Iptables y TCPWrappers en Linux:

Iptables es un *firewall* de línea de comandos de Linux que permite a los administradores de sistemas administrar el tráfico entrante y saliente a través de un conjunto de reglas de tabla configurables. Usa el *firewall* para filtrar el tráfico y permitir solo el necesario. Utilice también **TCPWrappers**, un sistema ACL de red basado en host para filtrar el acceso de red a Internet. Esto puede evitar muchos ataques de denegación de servicio con la ayuda de Iptables.



LECTURAS PARA REFORZAR LA UNIDAD



Capítulo 1

- Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (Fourth Edition). Pearson.

Capítulo 2

- Rendek, L. (2020, 2 mayo). *How to disable/enable SELinux on Ubuntu 20.04 Focal Fossa Linux*. Linux Tutorials - Learn Linux Configuration.
- Brown, K. (2020, 27 agosto). *The Beginner's Guide to iptables, the Linux Firewall*. How-To Geek. Tutorials Point.

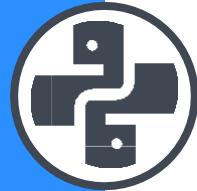


CONCLUSIÓN

Como pudimos comprender a lo largo de la presente unidad, reforzar el sistema operativo representa una prioridad para la seguridad de tu servidor, así como para la protección de los datos del usuario. En este sentido, para *adelantarse a los atacantes*, es importante establecer el sistema Linux con configuraciones personalizadas, que dificulten a los atacantes escanear el sistema y explotar vulnerabilidades comunes para agregar capas de seguridad a su servidor. Cabe señalar que es muy difícil garantizar al 100% la seguridad de un sistema; sin embargo, fortalecer el sistema operativo reduce en gran medida los riesgos asociados con los ataques cibernéticos.



FORO UNIDAD 1



Entorno de trabajo.



Participa en el foro enviando imágenes que demuestren que ya tienes acceso a las siguientes herramientas en su versión de prueba:

Sistema operativo Linux

Presiona el botón para participar en el foro.





¡FELICIDADES!

Acabas de concluir la **primera unidad** de tu curso *Sistemas Operativos II*. Te invitamos a finalizar este esfuerzo realizando el examen parcial correspondiente. Para ello, debes regresar a la pantalla principal y dar clic en *Presentar examen*.

UNIDAD 2

GESTIÓN DE LA CARGA DE TRABAJO





2.1

TEMARIO

2.2

SISTEMAS
DISTRIBUIDOS
Y CLUSTERS

PROCESAMIENTO
EN PARALELO
Y MULTITAREA



INTRODUCCIÓN

En esta segunda unidad aprenderás a identificar un sistema distribuido. Asimismo, comprenderás cómo trabajan los *cluster*, y cuál es la diferencia entre estos dos.

Además, también aprenderás a encontrar las diferencias entre un procesamiento paralelo y una multitarea, así como su funcionamiento básico y sus arquitecturas.

COMPETENCIAS A DESARROLLAR



- El alumno comprenderá de manera clara el funcionamiento básico tanto de los sistemas distribuidos como del *cluster*.

- El alumno estudiará el funcionamiento de un sistema de procesamiento paralelo, por una parte, y de los sistemas multitarea, por la otra.

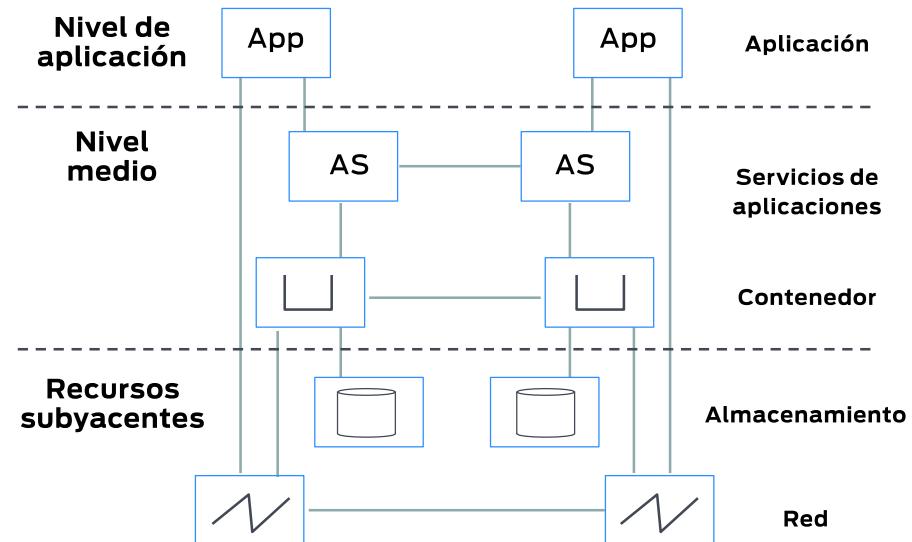
SISTEMAS DISTRIBUIDOS Y CLUSTERS

¿Qué es un Sistema Distribuido?

Es un entorno informático en el que varios componentes **se distribuyen en varias computadoras** u otros dispositivos informáticos en una red.

Estos dispositivos dividen el trabajo, coordinando sus esfuerzos para completarlo de manera más eficiente, a diferencia de lo que lograría un solo dispositivo.

Arquitectura de un sistema distribuido



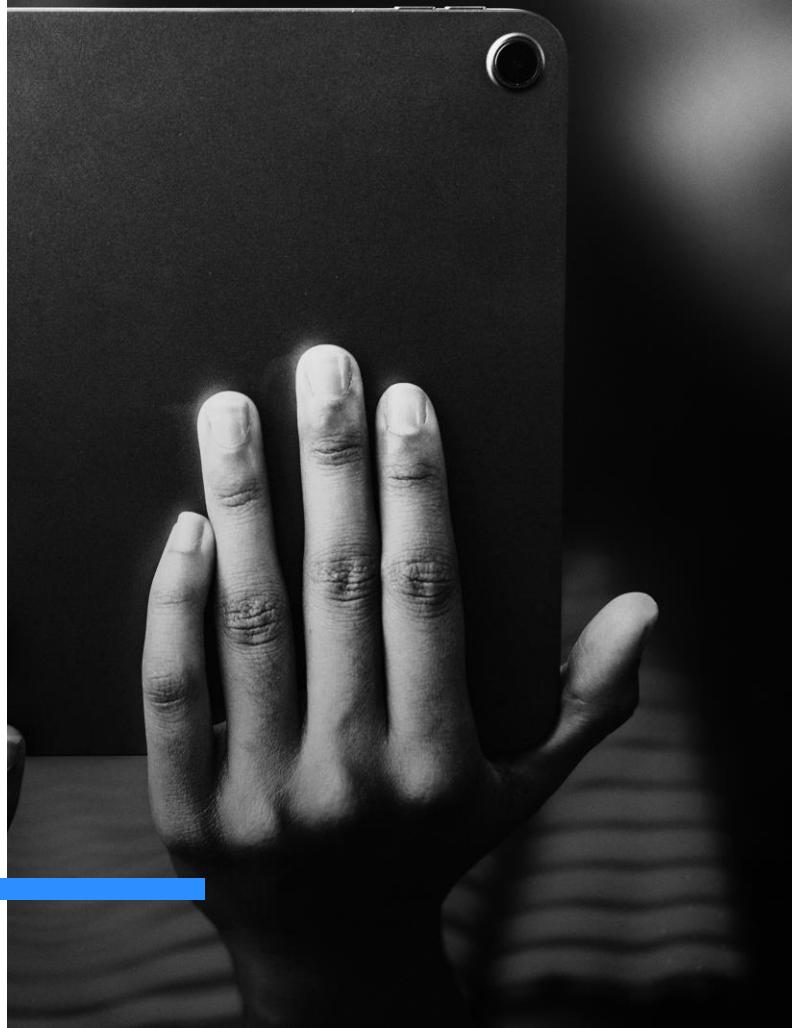


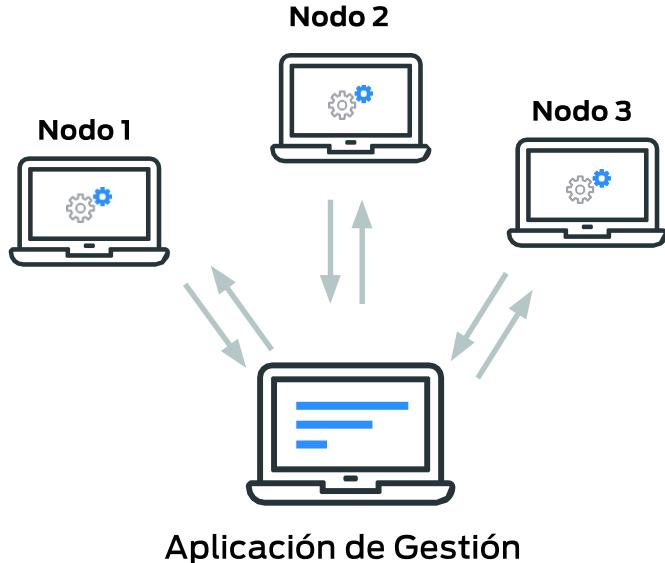
¿Cómo Funciona un Sistema Distribuido?

Los sistemas distribuidos han evolucionado con el tiempo, pero las implementaciones más comunes de hoy en día están diseñadas en gran medida para operar a través de Internet y, más específicamente, de la Nube.

Además, es importante saber que un sistema distribuido comienza con una tarea.

En este sentido, las aplicaciones distribuidas dividen el trabajo en partes. Por ejemplo, en un editor de video en un equipo cliente, el algoritmo entrega un cuadro del video a cada una de las docenas de computadoras (o **nodos**) diferentes para completar la representación requerida. Una vez que se completa el marco, la aplicación de administración le da al nodo un nuevo marco para trabajar. Este proceso continúa hasta que se termina el video y se vuelven a unir todas las piezas.





Cabe destacar que un sistema distribuido no tiene que detenerse en solo 2 o 3 nodos: el trabajo puede distribuirse entre cientos o incluso miles de nodos, realizando una tarea que podría haber tomado días en cuestión de minutos.

Características clave de un sistema distribuido:

- Escalabilidad
- Concurrency
- Disponibilidad/tolerancia a fallas
- Transparencia
- Heterogeneidad
- Replicación





- **Escalabilidad.** Es la capacidad de crecer a medida que aumenta el tamaño de la carga de trabajo. Se trata de una característica esencial de los sistemas distribuidos, que se logra agregando nodos adicionales a la red, según sea necesario.
- **Concurrencia.** Los componentes del sistema distribuido se ejecutan simultáneamente. También se caracterizan por la falta de un “reloj global”, cuando las tareas ocurren fuera de secuencia y a diferentes velocidades.

- **Disponibilidad/tolerancia a fallas.** Es decir, cuando un nodo falla, los nodos restantes pueden continuar operando sin interrumpir el cómputo general.
- **Transparencia.** Un programador externo o usuario final ve un sistema distribuido como una sola unidad computacional, en lugar de sus partes subyacentes. Esto permite a los usuarios interactuar con un solo dispositivo lógico en lugar de preocuparse por la arquitectura del sistema.





- 
- **Heterogeneidad.** En la mayoría de los sistemas distribuidos, los nodos y los componentes suelen ser asíncronos, con hardware, *middleware*, software y sistemas operativos diferentes. Esto permite ampliar los sistemas distribuidos con la adición de nuevos componentes.
 - **Replicación.** Los sistemas distribuidos permiten compartir información y mensajería, lo que garantiza la coherencia entre recursos redundantes, y mejora la tolerancia a fallas, la confiabilidad y la accesibilidad.

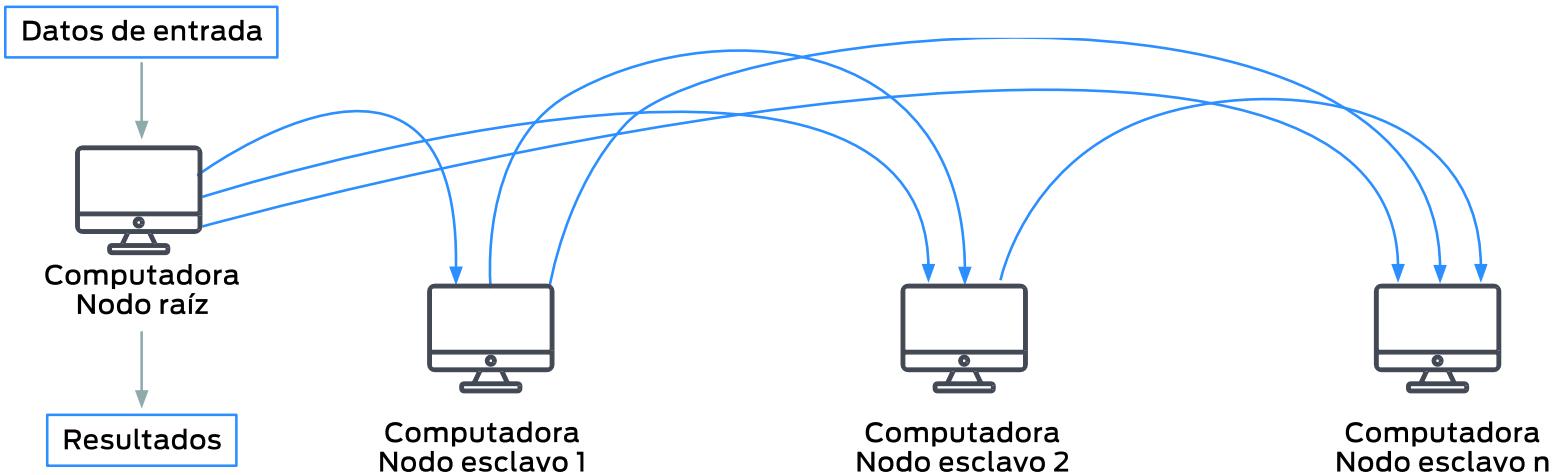
¿Qué es un *cluster*?:

Es un grupo de dos o más computadoras (nodos) que se ejecutan paralelamente para realizar una misma tarea. Lo anterior facilita que una gran cantidad de tareas individuales se distribuyan entre todos los nodos del *cluster*. En consecuencia, dichas tareas aprovechan la memoria conjunta, así como la potencia de procesamiento de cada nodo, lo que aumenta el rendimiento general del sistema.



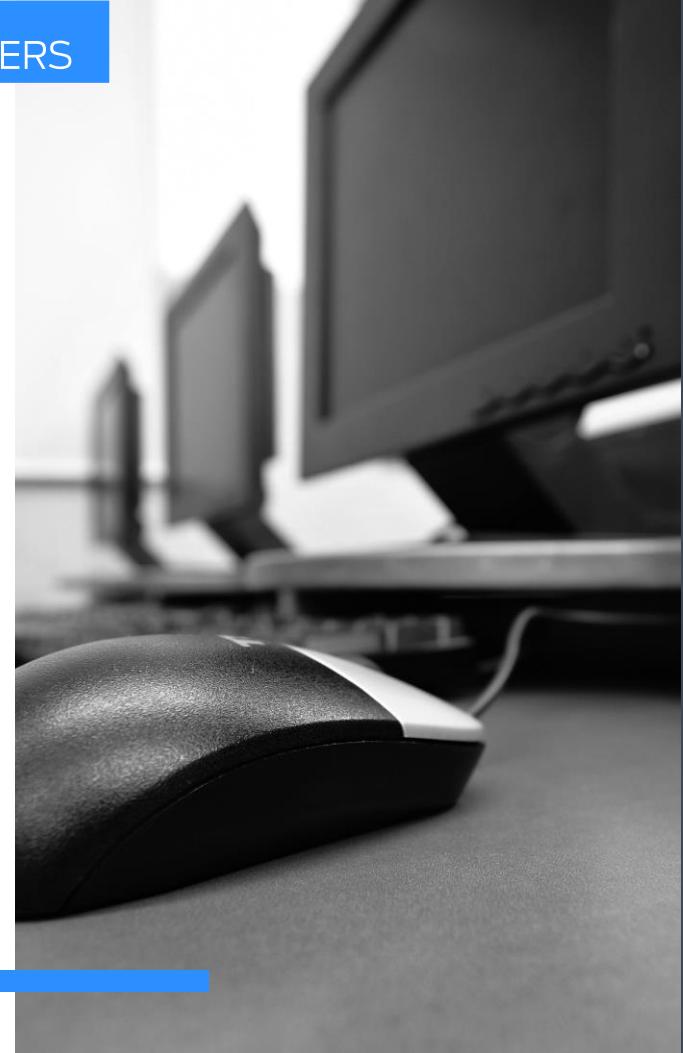
Para construir un *cluster* de computadoras:

- Cada nodo debe estar conectado en una misma red para permitir su intercomunicación.
- Se puede tener un dispositivo de almacenamiento compartido y/o almacenamiento local en cada nodo.
- Al menos un nodo se designa como **nodo líder** y actúa como punto de entrada al *cluster*. Este puede ser responsable de delegar el trabajo entrante a los otros nodos.



Idealmente, un *cluster* funciona como si fuera un solo sistema. Por lo tanto, cuando un usuario accede a este, no debería necesitar saber si el sistema es un *cluster* o una máquina individual. Además, un *cluster* debe diseñarse para **minimizar la latencia** y **evitar cuellos de botella** en la comunicación de nodo a nodo.

Los *cluster* de computadoras generalmente se pueden **clasificar en 3 tipos**:





Alta disponibilidad:

- **Disponibilidad.** Es la accesibilidad de un sistema o servicio durante un período de tiempo.
- **Resiliencia.** Es decir, qué tan bien se recupera un sistema de una falla.
- **Tolerancia a fallas.** Es la capacidad de un sistema para continuar brindando un servicio en caso de falla.
- **Confiabilidad.** Se requiere que un sistema funcione como se espera.
- **Redundancia.** Es la duplicación de recursos críticos para mejorar la confiabilidad del sistema.

2

► Equilibrio de carga:

Es el acto de distribuir el tráfico entre los nodos de un *cluster*, con la finalidad de optimizar el rendimiento y evitar así que un único nodo reciba una cantidad desproporcionada de trabajo.

Se puede instalar un equilibrador de carga en los nodos principales, o bien, aprovisionarlo por separado del *cluster*. Al realizar comprobaciones de estado en cada nodo del *cluster*, el equilibrador de carga puede detectar si un nodo ha fallado, en cuyo caso enrutará el tráfico entrante a los otros nodos.

3 ► Escalado

Hay **2 clasificaciones** de escalado:

- **Vertical.** También conocido como escalado hacia arriba, o hacia abajo. Busca aumentar o disminuir los recursos asignados a un proceso, como la cantidad de memoria, de núcleos de procesador o el almacenamiento disponible.
- **Horizontal.** Es cuando se ejecutan trabajos paralelos adicionales en el sistema.



► Rendimiento:

Cuando se trata de paralelización, los *cluster* pueden lograr niveles más altos que una sola máquina, ya que no están limitados por una determinada cantidad de núcleos de procesador u otro hardware.

Además, por otro lado, el escalado horizontal puede maximizar el rendimiento, debido a que evita que el sistema se quede sin recursos.



VIDEO

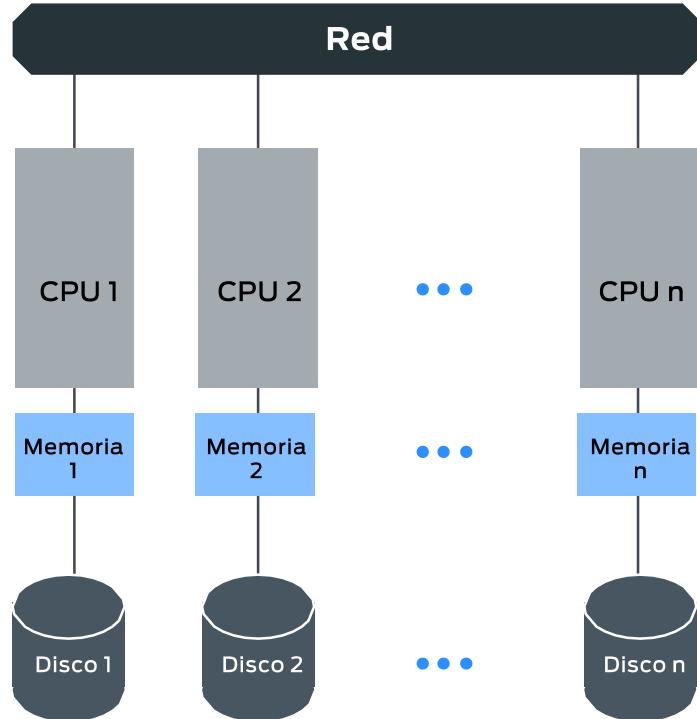
Te invitamos a ver el siguiente video:



PROCESAMIENTO EN PARALELO Y MULTITAREA

Procesamiento paralelo

Se refiere al concepto de acelerar la ejecución de un programa dividiendo el programa en múltiples fragmentos que pueden ejecutarse simultáneamente, cada uno en su propio procesador. Un programa que se ejecuta en “n” procesadores puede ejecutarse “n” veces más rápido que si se usara un solo procesador.



PROCESAMIENTO EN PARALELO Y MULTITAREA



Anteriormente, se otorgaban múltiples procesadores dentro de una *computadora paralela* diseñada para ello. Con esto en mente, ahora Linux admite sistemas SMP. En estos, múltiples procesadores comparten una sola interfaz de memoria y bus dentro de un solo nodo.

Además, también es posible que un grupo de computadoras esté interconectado por una red para formar un *cluster* de procesamiento paralelo.

Aunque el uso de múltiples procesadores puede acelerar muchas operaciones, la mayoría de las aplicaciones todavía no pueden beneficiarse del procesamiento paralelo. Por lo anterior, es importante conocer que **el procesamiento paralelo es apropiado solo si...**

- Tu aplicación tiene suficiente paralelismo para hacer un buen uso de varios procesadores.
- El programa de aplicación en cuestión ya se ha paralelizado.
- Estás interesado en investigar problemas relacionados con el procesamiento paralelo.

En consecuencia, descubrirás que el procesamiento paralelo con Linux puede generar un rendimiento de supercomputadora (por así decirlo) para algunos programas que realizan cálculos complejos u operan con grandes conjuntos de datos.

Como beneficio adicional, también es fácil usar un sistema Linux paralelo para otras cosas cuando este no está ocupado ejecutando un trabajo paralelo.



Terminología

Nombre	Descripción
SIMD	Flujo de instrucción única, flujo de datos múltiples. Es un modelo de ejecución en paralelo en el que todos los procesadores ejecutan la misma operación al mismo tiempo, pero cada procesador puede operar con sus propios datos.
MIMD	Flujo de instrucciones múltiples, flujo de datos múltiples. Es un modelo de ejecución en paralelo en el que cada procesador actúa esencialmente de forma independiente.
SPMD	Programa único, datos múltiples. Es una versión restringida de MIMD en la que todos los procesadores ejecutan el mismo programa. A diferencia de SIMD, cada procesador que ejecuta código SPMD puede tomar una ruta de flujo de control diferente a través del programa.

Nombre	Descripción
SMP	Symmetric Multi-Processor. Se refiere al concepto de sistema operativo de un grupo de procesadores que trabajan juntos como pares, de modo que cualquier trabajo puede ser realizado igualmente bien por cualquier procesador.
SWAR	SIMD dentro de un registro. Es un término genérico para el concepto de dividir un registro en varios campos enteros y usar operaciones de ancho de registro para realizar cálculos SIMD paralelos en esos campos. Dada una máquina con registros de k bits, rutas de datos y unidades de función, se sabe desde hace mucho tiempo que las operaciones de registro ordinarias pueden funcionar como operaciones SIMD paralelas en valores de campo de hasta n , k / n bits.



Multitarea. Se trata de un sistema operativo en el que múltiples procesos (tareas) pueden ejecutarse en una computadora aparentemente de manera simultánea y sin interferir entre sí. Es decir, cada tarea ofrece la ilusión de que es el único proceso en la computadora, y que tiene acceso exclusivo a todos los servicios del sistema operativo. Los procesos que se ejecutan simultáneamente pueden representar diferentes programas, diferentes partes de un solo programa y diferentes instancias de un solo programa.

La cantidad total de tareas que pueden ejecutarse en el sistema en cualquier momento depende de varios **factores**:

- Tamaño de la memoria
- Velocidad de la CPU
- Tamaño de los programas

Los procesos están completamente protegidos entre sí, al igual que el *Kernel* en un sistema bien diseñado está protegido de todos los procesos. Así, la interrupción del funcionamiento en un proceso no hará que otro programa o todo el sistema se bloquee. No obstante, los procesos se comunican entre sí cuando es necesario.

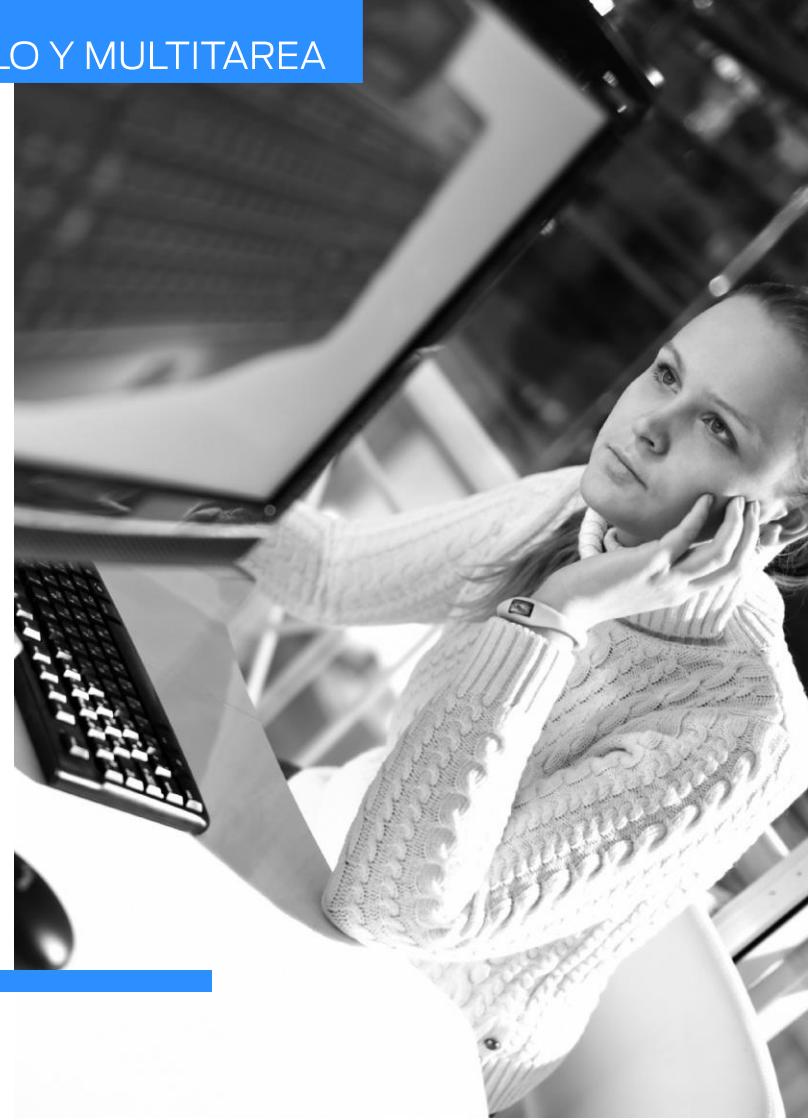


Multitarea en Linux. Una parte del *Kernel* (**planificador**) realiza un seguimiento de todos los programas en ejecución, para después asignar el tiempo del procesador, ejecutando varios programas de manera simultánea.

De esta manera, a cada proceso se le asigna su propia memoria, y el *Kernel* le prohíbe acceder a la memoria fuera de su área asignada.

Diferentes procesos comparten datos por medio de blocs de notas, comunes en la memoria. Esto en vez de escribir directamente en el espacio de memoria de los demás. En consecuencia, si un programa falla, ningún otro proceso se verá afectado.

El **reinicio no planificado** con GNU/Linux es extremadamente raro. Es normal que un sistema GNU/Linux funcione ininterrumpidamente durante semanas o meses sin reiniciar.





Existen múltiples tareas durante toda la duración de una sesión de Linux, incluso si no se están ejecutando programas de aplicación de usuario.

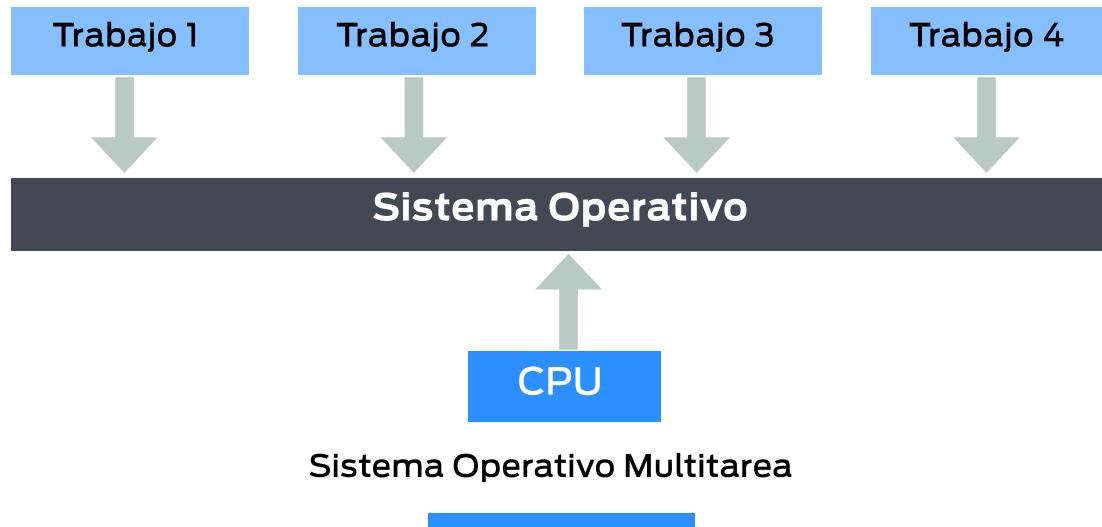
La única excepción es un período muy breve durante el arranque, cuando el comando `init`, que es el primer proceso que se crea en un sistema recién iniciado, aún no ha generado procesos adicionales.

El **tiempo compartido** (multitarea) es una extensión lógica de la **multiprogramación**. En este sentido, la CPU ejecuta varios trabajos, alternando entre ellos. Sin embargo, los cambios ocurren tan frecuentemente que los usuarios tienen la posibilidad de interactuar con cada programa mientras este se ejecuta.



Dadas las anteriores consideraciones, un sistema operativo multitarea permite ejecutar múltiples procesos al mismo tiempo. No obstante, estos no se ejecutan literalmente al mismo tiempo, ya que solo hay una CPU.

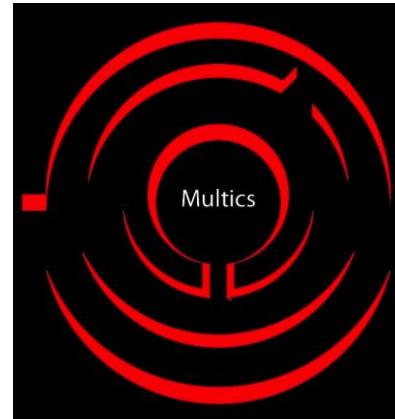
Lo que pasa es que un proceso se ejecuta durante un tiempo, y luego el sistema operativo interrumpe, almacena el estado del proceso actual, restaura el contexto de otro y permite que ese otro proceso se ejecute durante un tiempo.



UNIX y **MULTICS** son sistemas operativos multitarea. No obstante, estos funcionan con hardware más caro.

MS-DOS, en contraste, es un ejemplo de un sistema operativo no multitarea.

UNIX



LECTURAS PARA REFORZAR LA UNIDAD



Capítulo 1

- Milberg, K. (2007, 9 abril). *Linux clusters vs. grids*. SearchDataCenter.
- Kiarie, J. (2020, 25 septiembre). *10 Linux Distributions and Their Targeted Users*.



LECTURAS PARA REFORZAR LA UNIDAD



Capítulo 2

- Universidad Tecnológica Nacional Facultad Regional Santa Fe, Tschanz, R., & Smerling, L. (2001). *Procesamiento paralelo: qué tener en cuenta para aprovecharlo. Conceptos y alternativas en Linux.* (N.o 1).
- University of Paderborn, Beisel, T., Wiersema, T., Plessl, C., & Brinkman, A. (2020). *Cooperative Multitasking for Heterogeneous Accelerators in the Linux Completely Fair Scheduler.*



ACTIVIDAD INTEGRADORA 1



Te invitamos a realizar la siguiente actividad:

Presiona el botón para descargar la actividad:

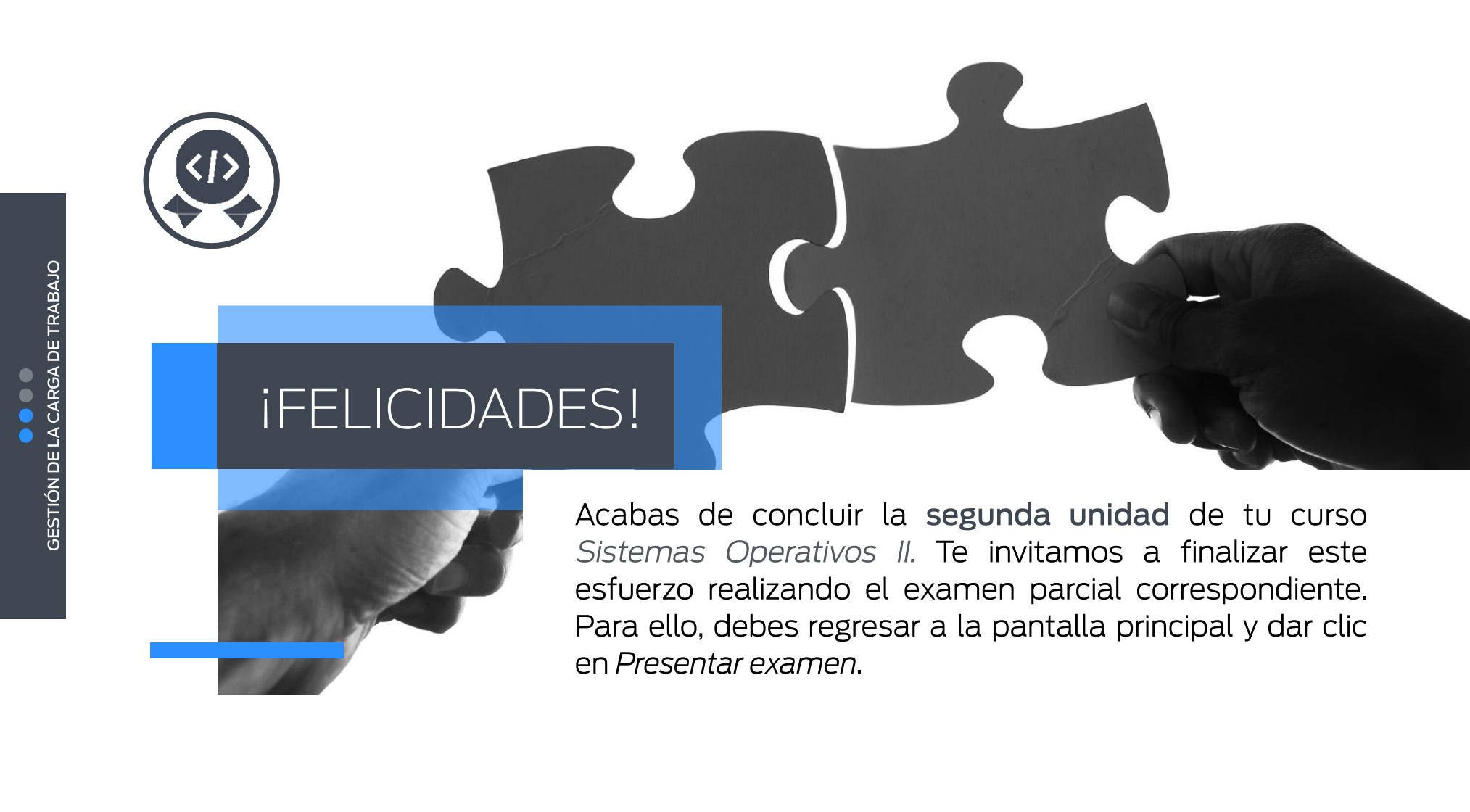


Presiona el botón para entregar la actividad:



CONCLUSIÓN

Las computadoras modernas son sistemas inherentemente distribuidos, y perdemos oportunidades para enfrentar los desafíos del sistema operativo del nuevo hardware si ignoramos los conocimientos de la investigación de sistemas distribuidos. En este sentido, hemos tratado de salir de la negación aplicando las ideas resultantes a una nueva arquitectura de sistema operativo, el *multikernel*.



¡FELICIDADES!

Acabas de concluir la **segunda unidad** de tu curso *Sistemas Operativos II*. Te invitamos a finalizar este esfuerzo realizando el examen parcial correspondiente. Para ello, debes regresar a la pantalla principal y dar clic en *Presentar examen*.

UNIDAD 3

RESOLUCIÓN DE PROBLEMAS





3.1

TEMARIO

3.2

RESOLUCIÓN
DE PROBLEMAS DE
DESEMPEÑO

RESOLUCIÓN
DE PROBLEMAS
DE RED



INTRODUCCIÓN

En esta tercera unidad estudiarás y comprenderás el funcionamiento de los comandos utilizados para resolver los distintos problemas de rendimiento del equipo con Linux.

Adicionalmente, conocerás los comandos para resolver problemas que se puedan presentar con la red.

COMPETENCIAS A DESARROLLAR



- El alumno estudiará y comprenderá los comandos para la resolución de problemas de rendimiento que se puedan presentar con Linux.

- El alumno estudiará y comprenderá los comandos necesarios para la resolución de problemas de red con Linux.

Según IBM, la **resolución de problemas** es:

“Un proceso que es a menudo iterativo, porque cuando se elimina un cuello de botella el rendimiento ahora está restringido por alguna otra parte del sistema. Por ejemplo, la sustitución de discos duros lentos por otros más rápidos podría cambiar el cuello de botella a la CPU de un sistema”.





Por lo tanto...

Los problemas de rendimiento ocurren en diferentes sistemas operativos o aplicaciones, y cada uno requiere un enfoque de resolución único.

La mayoría de los problemas están relacionados con la CPU, la memoria, las redes y la entrada/salida (E/S).

Cada una de estas áreas genera síntomas diferentes y requiere un diagnóstico y solución diferente.

Existen casos en los que los problemas de rendimiento pueden deberse a un **error de instalación/configuración** de la aplicación. Por ejemplo, que una aplicación web con capa de caché no esté configurada correctamente. En consecuencia, más solicitudes regresan al servidor de origen, en lugar de ser atendidas por el caché.

A continuación, algunos **comandos para solucionar problemas de este tipo en Linux**:





► Top:

- Es un comando de supervisión del rendimiento utilizado por muchos administradores de sistemas para supervisar el rendimiento de Linux.
- Se utiliza para mostrar todos los procesos activos y en ejecución en tiempo real en una lista ordenada, y se actualiza periódicamente.
- Muestra el uso del CPU, el uso de la memoria, la memoria de intercambio, el tamaño de la caché, el tamaño del búfer, el usuario, etcétera.

Además, Top muestra una alta utilización de la memoria y la CPU de los procesos en ejecución. Este comando es sumamente útil para que los administradores del sistema supervisen y tomen medidas correctivas en caso de necesitarse.

```
# top
```

```
top - 09:49:45 up 4:55, 4 users, load average: 0.01, 0.02, 0.00
Tasks: 133 total, 1 running, 132 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.3 hi, 0.3 si, 0.0 st
Mib Mem : 1070.8 total, 268.2 free, 253.5 used, 549.1 buff/cache
Mib Swap: 9216.0 total, 9174.0 free, 42.0 used. 648.0 avail Mem

          PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM TIME+ COMMAND
 37345 root      20   0  64512  4944  4064 R  0.3  0.5  0:00.07 top
    1 root      20   0 264232 10916  7052 S  0.0  1.0  0:05.32 systemd
    2 root      20   0      0    0    0 S  0.0  0.0  0:00.00 kthreadd
    3 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 rcu_gp
    4 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 rcu_par_gp
    6 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 kworker/0:0H-kblockd
    8 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 mm_percpu_wq
    9 root      20   0      0    0    0 S  0.0  0.0  0:00.86 ksoftirqd/0
   10 root      20   0      0    0    0 I  0.0  0.0  0:00.76 rcu_sched
   11 root      rt   0      0    0    0 S  0.0  0.0  0:00.00 migration/0
   12 root      rt   0      0    0    0 S  0.0  0.0  0:00.02 watchdog/0
   13 root      20   0      0    0    0 S  0.0  0.0  0:00.00 cpuhp/0
   15 root      20   0      0    0    0 S  0.0  0.0  0:00.00 kdevtmpfs
   16 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 netns
   17 root      20   0      0    0    0 S  0.0  0.0  0:00.00 kaudited
   18 root      20   0      0    0    0 S  0.0  0.0  0:00.00 khungtaskd
   19 root      20   0      0    0    0 S  0.0  0.0  0:00.00 oom_reaper
   20 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 writeback
   21 root      20   0      0    0    0 S  0.0  0.0  0:00.00 kcompactd0
   22 root      25   5      0    0    0 S  0.0  0.0  0:00.00 ksmd
   23 root      39  19      0    0    0 S  0.0  0.0  0:00.37 khugepaged
   24 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 crypto
   25 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 integrityd
   26 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 kblockd
   27 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 tpm_dev_wq
   28 root      0 -20      0    0    0 I  0.0  0.0  0:00.00 md
   29 root      0 -20      0    0    0 T  0.0  0.0  0:00.00 edac_poller
```

► VmStat:

- Este comando se utiliza para mostrar estadísticas de memoria virtual, subprocesos del núcleo, discos, procesos del sistema, bloques de E/S, interrupciones, actividad de la CPU y mucho más.
- Por defecto, no está disponible en los sistemas Linux. Es necesario instalar un paquete llamado [sysstat](#) que incluye un programa [vmstat](#).

```
$ sudo yum install sysstat      [On Older CentOS/RHEL & Fedora]
$ sudo dnf install sysstat      [On CentOS/RHEL/Fedora/Rocky Linux & AlmaLinux]
$ sudo apt-get install sysstat  [On Debian/Ubuntu & Mint]
$ sudo pacman -S sysstat       [On Arch Linux]
```

El uso común del formato de comando [vmstat](#) se ve en la siguiente imagen:

```
[root@tecmint:~]# vmstat
procs -----memory----- swap-- io---- system-- cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
1 0 43008 275212 1152 561208 4 16 100 105 65 113 0 1 96 3 0
[root@tecmint:~]#
```

► Lsof:

- Este comando se usa para mostrar una lista de todos los archivos abiertos y los procesos.
- Uno de los principales usos para este comando es cuando un disco no se puede desmontar y se muestra el error de que los archivos se están usando o abriendo, ya que con este comando puedes identificar fácilmente qué archivos están en uso.



El formato más común para el comando `lsof` es:

COMMAND	PID	TID	TASKCMD	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
systemd	1			root	cwd	DIR	8,2	224	128	/
systemd	1			root	rtd	DIR	8,2	224	128	/
systemd	1			root	txt	REG	8,2	1567768	134930842	/usr/lib/systemd/systemd
systemd	1			root	mem	REG	8,2	2714928	134261052	/usr/lib64/libm-2.28.so
systemd	1			root	mem	REG	8,2	628592	134910905	/usr/lib64/libudev.so.1.6.11
systemd	1			root	mem	REG	8,2	969832	134261204	/usr/lib64/libsepolicy.so.1
systemd	1			root	mem	REG	8,2	1805368	134275205	/usr/lib64/libunistring.so.2.1.0
systemd	1			root	mem	REG	8,2	355456	134275293	/usr/lib64/libpcap.so.1.9.0
systemd	1			root	mem	REG	8,2	145984	134261219	/usr/lib64/libgpg-error.so.0.24.2
systemd	1			root	mem	REG	8,2	71528	134270542	/usr/lib64/libjson-c.so.4.0.0
systemd	1			root	mem	REG	8,2	371736	134910992	/usr/lib64/libdevmapper.so.1.02
systemd	1			root	mem	REG	8,2	26704	134275177	/usr/lib64/libattr.so.1.1.2448
systemd	1			root	mem	REG	8,2	3058736	134919279	/usr/lib64/libcrypto.so.1.1.1c
systemd	1			root	mem	REG	8,2	615504	134919281	/usr/lib64/libssl.so.1.1.1c
systemd	1			root	mem	REG	8,2	95232	134261206	/usr/lib64/libz.so.1.2.11
systemd	1			root	mem	REG	8,2	24736	134266276	/usr/lib64/libcap-ng.so.0.0.0
systemd	1			root	mem	REG	8,2	33224	134261224	/usr/lib64/libuuid.so.1.3.0
systemd	1			root	mem	REG	8,2	36824	134261050	/usr/lib64/libdl-2.28.so
systemd	1			root	mem	REG	8,2	553480	134260151	/usr/lib64/libpcre2-8.so.0.7.1
systemd	1			root	mem	REG	8,2	339328	134910989	/usr/lib64/libblkid.so.1.1.0
systemd	1			root	mem	REG	8,2	96056	134275191	/usr/lib64/liblz4.so.1.8.1
systemd	1			root	mem	REG	8,2	192024	134266238	/usr/lib64/liblzma.so.5.2.4
systemd	1			root	mem	REG	8,2	165624	134275210	/usr/lib64/libidn2.so.0.3.6
systemd	1			root	mem	REG	8,2	33240	134275311	/usr/lib64/libip4tc.so.0.1.0
						REG	8,2	145620	134266225	/usr/lib64/liblz4.so.1.8.1

Lista de archivos abiertos en Linux



► Htop:

- Es una herramienta de monitoreo de procesos de Linux muy avanzada, interactiva y en tiempo real.
- Cuenta con una interfaz fácil de usar para administrar procesos, teclas de acceso directo, vistas verticales y horizontales de los procesos, y mucho más.

```
CPU[|          0.7%]  Tasks: 53, 44 thr; 1 running
Mem[||||||| 237M/1.05G] Load average: 0.37 0.11 0.04
Swp[|        42.0M/9.00G] Uptime: 05:41:34

 PID USER PRI  NI   VIRT   RES   SHR S CPU% MEM% TIME+  Command
 38228 root  20   0 28020  4064 3352 R  0.7  0.4  0:00.09 htop
    1 root  20   0  258M 10920 7056 S  0.0  1.0  0:05.41 /usr/lib/system
   562 root  20   0  92964  8588 7748 S  0.0  0.8  0:01.14 /usr/lib/system
   599 root  20   0  116M  8980 5536 S  0.0  0.8  0:00.72 /usr/lib/system
   716 rpc   20   0  67120  4616 4280 S  0.0  0.4  0:00.05 /usr/bin/rpcbind
   726 root  16  -4  139M 2284 1880 S  0.0  0.2  0:00.08 /sbin/auditd
   727 root  16  -4  139M 2284 1880 S  0.0  0.2  0:00.00 /sbin/auditd
   728 root  16  -4  48484  2136 1832 S  0.0  0.2  0:00.03 /usr/sbin/sedis
   729 root  16  -4  139M 2284 1880 S  0.0  0.2  0:00.01 /sbin/auditd
   751 root  20   0  83672  5912 5048 S  0.0  0.5  0:00.43 /usr/lib/system
   755 avahi  20   0  82752  3412 3188 S  0.0  0.3  0:00.23 avahi-daemon: r
   756 root  20   0  207M  8508 8104 S  0.0  0.8  0:00.16 /usr/sbin/sssd
   760 root  20   0  26368  3692 3168 S  0.0  0.3  0:00.05 /usr/sbin/smart
   762 root  20   0  451M  7484 6088 S  0.0  0.7  0:00.09 /usr/sbin/Modem
   765 polkitd 20   0 1599M 19936 13248 S  0.0  1.8  0:03.59 /usr/lib/polkit
   766 dbus   20   0  91772  7692 4940 S  0.0  0.7  0:02.89 /usr/bin/dbus-d
   769 chrony 20   0  125M  2884 2460 S  0.0  0.3  0:00.22 /usr/sbin/chrony
   772 libstorag 20   0 18872  1908 1780 S  0.0  0.2  0:00.13 /usr/bin/lsmd -
   773 root   20   0 17408  1948 1800 S  0.0  0.2  0:00.00 /usr/sbin/mcelog
   775 rngd   20   0  156M  5812 5312 S  0.0  0.5  0:07.22 /sbin/rngd -f -
   779 root   20   0 25244  1960 1668 S  0.0  0.2  0:00.58 /bin/bash /usr/
   784 root   20   0  99M  2704 2336 S  0.0  0.2  0:00.00 /usr/sbin/gsspr
   785 root   20   0  451M  7484 6088 S  0.0  0.7  0:00.00 /usr/sbin/Modem
   786 root   20   0  99M  2704 2336 S  0.0  0.2  0:00.00 /usr/sbin/gsspr
   787 root   20   0  99M  2704 2336 S  0.0  0.2  0:00.00 /usr/sbin/gsspr

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
```

► Iotop:

- Es muy similar al comando superior y al programa [htop](#), pero con una función de contabilidad para monitorear y mostrar procesos y E/S de disco en tiempo real.
- Es muy útil para encontrar el proceso exacto y las lecturas/escrituras de disco de alto uso de los procesos.
- De forma predeterminada, no está disponible en Linux, por lo que debe instalarse como se muestra:

```
$ sudo yum install iotop      [On Older CentOS/RHEL & Fedora]
$ sudo dnf install iotop      [On CentOS/RHEL/Fedora/Rocky Linux & AlmaLinux]
$ sudo apt-get install iotop  [On Debian/Ubuntu & Mint]
$ sudo pacman -S iotop       [On Arch Linux]
```

El uso común del formato de comando iotop es:

Total DISK READ :			0.00 B/s	Total DISK WRITE :	0.00 B/s		
Actual DISK READ:			0.00 B/s	Actual DISK WRITE:	0.00 B/s		
TID	PRI0	USER	DISK READ	DISK WRITE	SWAPIN	IO>	COMMAND
1	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	systemd --switched-root
2	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kthreadd]
3	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_gp]
4	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_par_gp]
6	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kworker/0:0H-kblockd]
8	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[mm_percpu_wq]
9	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ksoftirqd/0]
10	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[rcu_sched]
11	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/0]
12	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[watchdog/0]
13	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[cpuhp/0]
15	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kdevtmpfs]
16	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[netns]
17	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kaudittd]
18	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[khungtaskd]
19	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[oom_reaper]
20	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[writeback]
21	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kcompactd0]
22	be/5	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ksmd]
23	be/7	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[khugepaged]
24	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[crypto]
25	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kintegrityd]
26	be/0	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kblockd]

iotop – Supervisar el uso de E/S de disco de Linux

► Iostat

- Recopila y muestra estadísticas de dispositivos de almacenamiento de entrada y salida del sistema.
- Se usa a menudo para rastrear problemas de rendimiento de dispositivos de almacenamiento.
- Para obtener el comando **iostat**, debes instalar un paquete llamado **sysstat**, como se muestra a continuación:

```
$ sudo yum install sysstat      [On Older CentOS/RHEL & Fedora]
$ sudo dnf install sysstat      [On CentOS/RHEL/Fedora/Rocky Linux & AlmaLinux]
$ sudo apt-get install sysstat  [On Debian/Ubuntu & Mint]
$ sudo pacman -S sysstat       [On Arch Linux]
```

```
[root@tecmint:~]# iostat
Linux 4.18.0-193.el8.x86_64 (tecmint)  04/05/2021      _x86_64_
avg-cpu:  %user   %nice  %system  %iowait  %steal   %idle
          0.21    0.03    0.59    2.50    0.00   96.67

Device      tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda       3.95      83.35      89.63  1782431    1916653
```

iostat – Supervisar las estadísticas de E/S del disco

► Psacct o Acct:

- Muy útil para monitorear la actividad de cada usuario en el sistema.
- Los *daemons* se ejecutan en segundo plano y vigilan de cerca la actividad general de cada usuario en el sistema, así como los recursos que están consumiendo.

```
[root@tecmint:~]# ac
      total      18.77
[root@tecmint:~]# ac -d
Jul  4  total      0.24
Today  total      18.54
[root@tecmint:~]# ac -p
      osboxes      6.26
      tecmint      2.47
      root      10.05
      total      18.78
[root@tecmint:~]# ac tecmint
      total      2.47
[root@tecmint:~]# ac -d tecmint
Today  total      2.47
[root@tecmint:~]#
```

psacct – Supervisar las actividades de los usuarios de Linux



VIDEO

Te invitamos a ver el siguiente video:

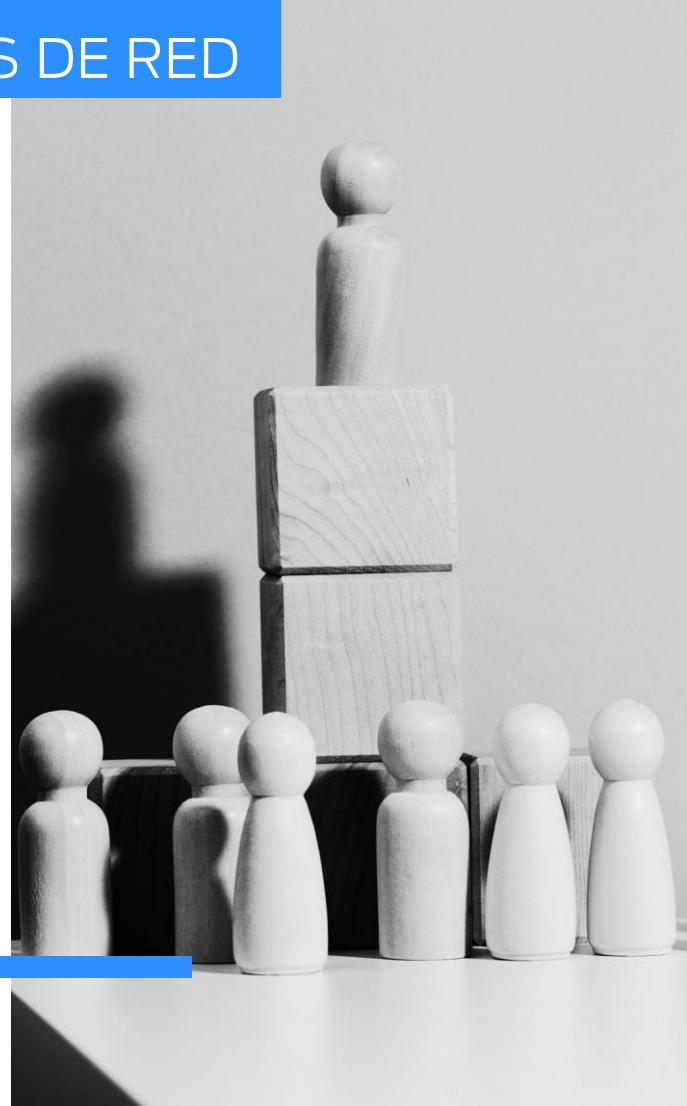


RESOLUCIÓN DE PROBLEMAS DE RED

La **resolución de problemas de red** tiene que ver con las medidas y los procesos colectivos que se utilizan para identificar, diagnosticar y resolver problemas, así como diversas cuestiones dentro de una red informática.

Se trata de un proceso sistemático cuyo objetivo es resolver problemas y restaurar las operaciones normales de la red.

Cabe señalar que la solución de problemas de red se realiza, principalmente, por ingenieros o administradores de red. Por lo general, se realiza para recuperar y establecer conexiones de red o de Internet en nodos/dispositivos finales.



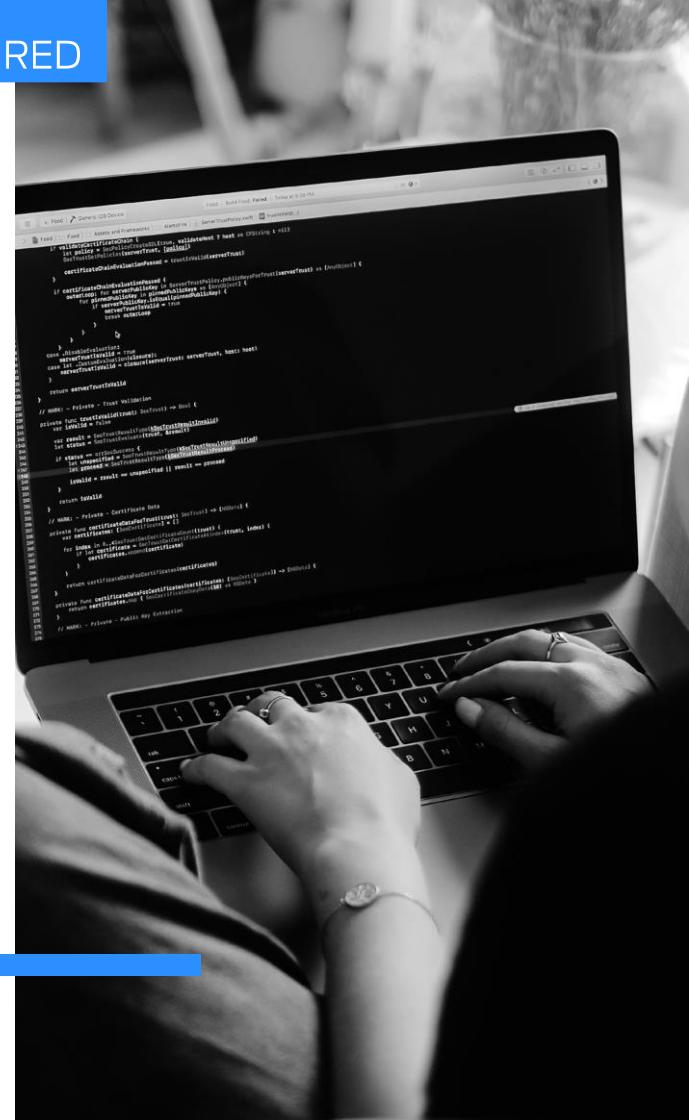


Algunos **procesos** dentro de la solución de problemas de red son:

- Encontrar y resolver problemas y establecer conexión a Internet.
- Configuración de enrutador, conmutador o cualquier dispositivo de *admin* de red.
- Instalación de cables o dispositivos wifi.
- Actualización de dispositivos de *firmware* en el conmutador del enrutador.
- Eliminación de virus.
- Adición, configuración y reinstalación de una impresora de red, entre otros.

Las responsabilidades rutinarias de un administrador de red, como la administración, el monitoreo, la configuración y la resolución de problemas **no requieren que aprenda herramientas complicadas de terceros**. En consecuencia, puede realizar todas estas tareas con herramientas preinstaladas en la mayoría de las distribuciones de Linux.

A continuación, algunos **comandos que utiliza un administrador de red de Linux**:



```
//base case 1, I have found a path!
pathorniiles.add(current);
return true;
}
else if(airlinesvisited.contains(current))
// base case 2, I have already been here
// and I am in a cycle
return false;
else
// I have not been here and it isn't
// the goal so check its partners
// and add them to the list
airlinesvisited.add(current);
// add this to the path
pathorniiles.add(current);
// find this airline in the network
int pos = -1;
int index = 0;
while(pos == -1 && index < network.size()){
    if(network.get(index).getname().equals(current))
        pos = index;
    index++;
}
//if not in the network, no partners
if(pos == -1)
    return false;
// loop through partners
index = 0;
String[] partners = network.get(pos).getpartners();
boolean foundPath = false;
while( !foundPath && index < partners.length()){
    String partner = partners[index];
    if( partner.equals(goal) || pathorniiles.contains(partner) )
        foundPath = true;
    index++;
}
if( !foundPath )
    pathorniiles.remove(pathorniiles.size() - 1);
return foundPath;
}

static class Airline{
    String name;
    String[] partners;
    static ArrayList<String> partners;
}

if( data == null || data.length > 0 )
    if( AirlineString[0] == data[0] )
        assert data != null && data.length > 0 : "Failed precondition";
    par
```



► Ifconfig:

- Es una utilidad de línea de comandos conocida para la configuración de la interfaz en los sistemas operativos Linux/Unix.
- Se utiliza para consultar y administrar los parámetros de la interfaz con la ayuda de *scripts* de configuración.
- Ayuda a habilitar o deshabilitar una interfaz de red.
- Permite asignar una dirección IP y una máscara de red a la interfaz seleccionada. Así, puedes ver las interfaces disponibles, direcciones IP, las direcciones de hardware y el tamaño máximo de la unidad de transmisión de mensajes enviados y recibidos con el tiempo que tarda un paquete en llegar a su destino.

Puedes **activar/desactivar** cualquier interfaz utilizando los parámetros arriba/abajo, de la siguiente manera:

```
sudo ifconfig up eth0  
sudo ifconfig down eth0
```

Para **asignar una dirección IP a una interfaz**:

```
sudo ifconfig eth0 192.168.120.5 netmask 255.255.255.0
```

Esta utilidad no está disponible en las distribuciones de Linux, y es posible que reciba un error “ [ifconfig: comando no encontrado](#) ”. Puede resolver el problema instalando el paquete [net-tools](#), con ayuda del administrador de paquetes de su distribución:

```
sudo apt-get install net-tools
```



► ip:

- Es una alternativa a **ifconfig**. No obstante, el alcance de su funcionalidad cubre dos capas del protocolo TCP/IP: la capa de enlace de datos y la capa de red.
- Muestra interfaces de red y configura sus dispositivos, al igual que la utilidad **ifconfig**.
- Muestra y modifica las tablas de enrutamiento del *Kernel* con la adición/eliminación de entradas de caché ARP.

► Ping:

- Ayuda a identificar la disponibilidad de una red y un *host*, ya que comprueba si se puede acceder a este o si se está ejecutando un servicio.
- Puede verificar problemas de conectividad de red, como alta latencia o caída de paquetes.
- Envía mensajes de solicitud de eco ICMP y espera los paquetes de respuesta eco ICMP para verificar la disponibilidad del host. La salida contiene el total de mensajes enviados y recibidos con el tiempo que tarda un paquete en llegar a su destino.



▶ Host:

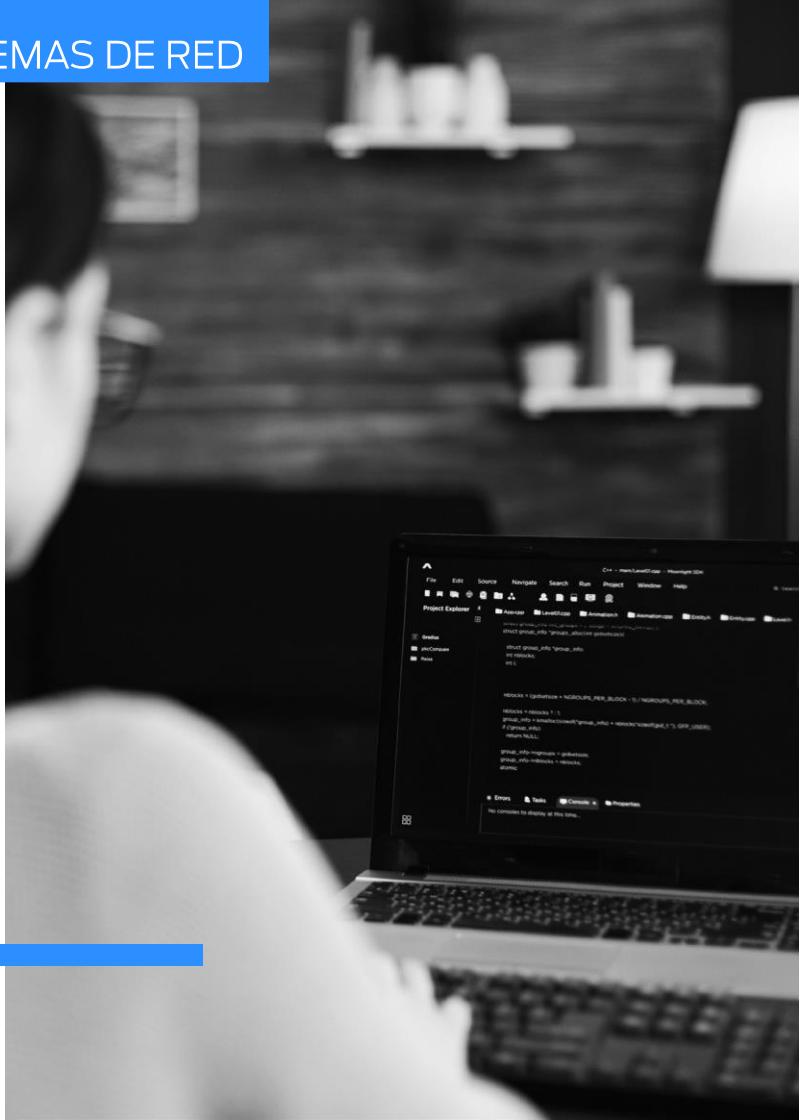
- Realiza búsquedas del DNS y resuelve el nombre de *host* en direcciones IP y viceversa.
- Soluciona los problemas del servidor DNS.
- Muestra y verifica los tipos de registro DNS NS y MX y los servidores DNS del ISP.

Por **ejemplo**, para encontrar NX para el sitio web de Google puedes utilizar este comando:

```
host -t ns google.com
```

► **Tcpdump:**

- Se utiliza para capturar o filtrar paquetes TCP/IP que se reciben o transfieren en una interfaz específica a través de una red.
 - Proporciona una opción para guardar los paquetes capturados en un archivo para su posterior análisis.
 - Está casi disponible en todas las principales distribuciones de Linux.



Funcionamiento de `tcpdump` como analizador de paquetes de red:

```
[root@tecmint:~]# tcpdump -i enp0s3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
10:21:05.119045 IP 192.168.0.124.45611 > tecmint.ssh: Flags [.], ack 2840833288, win 11768
10:21:05.128704 IP tecmint.47556 > _gateway.domain: 13517+ PTR? 159.0.168.192.in-addr.arpa
10:21:05.128986 IP tecmint.ssh > 192.168.0.124.45611: Flags [P.], seq 1:209, ack 0, win 14
10:21:05.129152 IP 192.168.0.124.45611 > tecmint.ssh: Flags [.], ack 209, win 11768, option
10:21:05.134483 IP _gateway.domain > tecmint.47556: 13517 NXDomain 0/1/0 (93)
10:21:05.135404 IP tecmint.47761 > _gateway.domain: 3697+ PTR? 124.0.168.192.in-addr.arpa.
10:21:05.153782 IP _gateway.domain > tecmint.47761: 3697 NXDomain 0/1/0 (93)
10:21:05.154517 IP tecmint.52148 > _gateway.domain: 14980+ PTR? 1.0.168.192.in-addr.arpa.
10:21:05.154762 IP tecmint.ssh > 192.168.0.124.45611: Flags [P.], seq 209:417, ack 0, win
10:21:05.154948 IP 192.168.0.124.45611 > tecmint.ssh: Flags [.], ack 417, win 11768, option
10:21:05.157361 IP _gateway.domain > tecmint.52148: 14980 NXDomain 0/1/0 (91)
10:21:05.158060 IP tecmint.ssh > 192.168.0.124.45611: Flags [P.], seq 417:1617, ack 0, win
10:21:05.158401 IP 192.168.0.124.45611 > tecmint.ssh: Flags [.], ack 1617, win 11766, option
10:21:05.158623 IP tecmint.ssh > 192.168.0.124.45611: Flags [P.], seq 1617:1985, ack 0, wi
10:21:05.158823 IP 192.168.0.124.45611 > tecmint.ssh: Flags [.], ack 1985, win 11768, option
10:21:05.159003 IP tecmint.ssh > 192.168.0.124.45611: Flags [P.], seq 1985:2353, ack 0, wi
10:21:05.159138 IP 192.168.0.124.45611 > tecmint.ssh: Flags [.], ack 2353, win 11768, option
```

► Netstat:

- Es una herramienta de línea de comandos para monitorear estadísticas de paquetes de red entrantes y salientes, así como estadísticas de interfaz.
- Es una herramienta muy útil para que todos los administradores de sistemas supervisen el rendimiento de la red y resuelvan problemas relacionados con esta.

```
[root@tecmint:~]# netstat -a | more
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp     0      0 0.0.0.0:sunrpc          0.0.0.0:*
tcp     0      0 tecmint:domain          0.0.0.0:*
tcp     0      0 0.0.0.0:ssh           0.0.0.0:*
tcp     0      0 localhost:postgres      0.0.0.0:*
tcp     0      0 tecmint:ssh            192.168.0.124:45611  ESTABLISHED
tcp6    0      0 [::]:sunrpc           [::]:*
tcp6    0      0 [::]:ssh              [::]:*
tcp6    0      0 localhost:postgres      [::]:*
udp     0      0 0.0.0.0:mdns           0.0.0.0:*
udp     0      0 localhost:323          0.0.0.0:*
udp     0      0 tecmint:domain          0.0.0.0:*
udp     0      0 0.0.0.0:bootps          0.0.0.0:*
udp     0      0 tecmint:bootpc          _gateway:bootps        ESTABLISHED
etc...
```

► IPTraf:

- Es una utilidad de monitoreo de red en tiempo real basada en consola.
 - Recopila una variedad de información, como el monitor de tráfico IP que pasa por la red, los detalles de ICMP, los desgloses de tráfico TCP/UDP, el paquete de conexión TCP y los recuentos de bytes.
 - Además, recopila información de estadísticas generales y detalladas de la interfaz de TCP, UDP, IP, ICMP, no IP, etcétera.

► NetHogs:

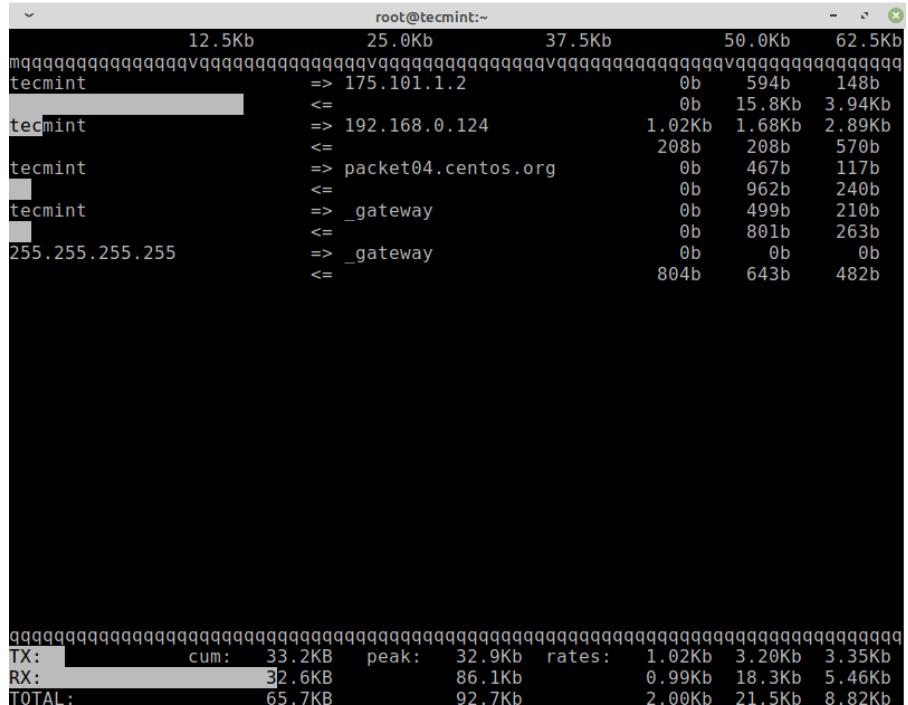
- Es similar al comando top de Linux, que mantiene una pestaña en cada actividad de red de proceso en su sistema.
- Realiza un seguimiento del ancho de banda del tráfico de red en tiempo real utilizado por cada programa o aplicación.

NetHogs version 0.8.5

PID	USER	PROGRAM	DEV	SENT	RECEIVED
37256	root	sshd: root@pts/0	enp0s3	0.180	0.064 KB/sec
?	root	..2.168.0.159:2812-192.16		0.000	0.000 KB/sec
?	root	unknown TCP		0.000	0.000 KB/sec
TOTAL				0.180	0.064 KB/sec

► iftop:

- Muestra una lista actualizada con frecuencia de la utilización del ancho de banda de la red que pasa a través de la interfaz de red en tu sistema.
 - Se considera para el uso de la red, lo que **top** hace para el uso de la CPU.
 - Monitorea una interfaz seleccionada y muestra el uso actual del ancho de banda entre dos hosts.



► Arpwatch:

- Es un tipo de programa que está diseñado para monitorear la resolución de direcciones del tráfico de la red Ethernet en una red Linux.
- Vigila continuamente el tráfico de Ethernet y produce un registro de los cambios de pares de direcciones IP y MAC junto con una marca de tiempo en una red.
- Tiene una función para enviar alertas por correo electrónico a los administradores, cuando se agrega o cambia un emparejamiento.
- Es muy útil para detectar la suplantación de ARP en una red.

RESOLUCIÓN DE PROBLEMAS DE RED

```
[root@tecmint:~]# arpwatch -i enp0s3
[root@tecmint:~]# tail -f /var/log/messages
Apr  5 11:40:42 osboxes systemd[1]: Started Network Manager Script Dispatcher Service.
Apr  5 11:42:33 osboxes systemd[1]: Started /usr/bin/systemctl start man-db-cache-update.
Apr  5 11:42:33 osboxes systemd[1]: Starting man-db-cache-update.service...
Apr  5 11:42:33 osboxes systemd[1]: Reloading.
Apr  5 11:42:48 osboxes kernel: device enp0s3 entered promiscuous mode
Apr  5 11:42:48 osboxes arpwatch[41055]: listening on enp0s3
Apr  5 11:42:52 osboxes systemd[1]: Started man-db-cache-update.service.
Apr  5 11:43:00 osboxes arpwatch[41055]: new station 192.168.0.1 98:da:c4:3e:c8:4c
Apr  5 11:43:00 osboxes arpwatch[41055]: new station 192.168.0.124 38:b1:db:7c:78:c7
Apr  5 11:43:05 osboxes arpwatch[41055]: new station 192.168.0.159 08:00:27:d2:f
```

Arpwatch – Monitorea el tráfico ARP

LECTURAS PARA REFORZAR LA UNIDAD



Capítulo 1

- A. (2020b, diciembre 4). *How to Troubleshoot Performance Issues on Linux – Beginners Guide*. The Geek Diary.
- D. (2021a, junio 25). *Resolución de problemas de rendimiento*. IBM.



LECTURAS PARA REFORZAR LA UNIDAD



Capítulo 2

- Ruostemaa, J. (2022, 17 mayo). *How to troubleshoot network connectivity with Linux server*. UpCloud.
- Welekwe, A. (2021, 3 agosto). *10 Best Linux Network Troubleshooting Tools for 2022*. Comparitech.



ACTIVIDAD INTEGRADORA 2



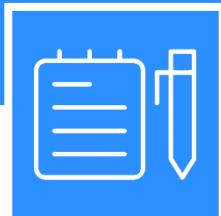
Te invitamos a realizar la siguiente actividad:



Presiona el botón para descargar la actividad:



Presiona el botón para entregar la actividad:



CONCLUSIÓN

La resolución de problemas de rendimiento se basa en la comparación. Para saber que el rendimiento de un equipo es malo, se debe tener un punto de referencia preexistente de cuando el rendimiento era bueno. Aunque se puede continuar dando cientos de ejemplos de problemas relacionados con el rendimiento, un mejor enfoque es comprender las herramientas y practicar con ellas de manera continua.

Esto también aplica para los problemas con la red, ya que, cuando surge un problema, lo primero que se debe ser hacer es evaluarlo para determinar de qué se trata realmente. Lo anterior es importante para determinar la naturaleza específica del problema para poder encontrar, a su vez, la solución particular y adecuada.



¡FELICIDADES!

Acabas de concluir la **tercera unidad** de tu curso *Sistemas Operativos II*. Te invitamos a finalizar este esfuerzo realizando el examen parcial correspondiente. Para ello, debes regresar a la pantalla principal y dar clic en *Presentar examen*.



UNIDAD 4

GESTIÓN DE ENTORNOS EXTERNOS





4.1

TEMARIO

4.2

ALMACENAMIENTO
EN AWS

DESPLIEGUE
DE APLICACIONES
CON DEVOPS



INTRODUCCIÓN

En esta última unidad de la materia *Sistemas Operativos II* aprenderás acerca de Amazon Linux, un entorno de ejecución que nos permite a los desarrolladores realizar ejecuciones de nuestras aplicaciones optimizadas.

Además, aprenderás sobre la orquestación de aplicaciones de la mano con DevOps y Linux.



COMPETENCIAS A DESARROLLAR

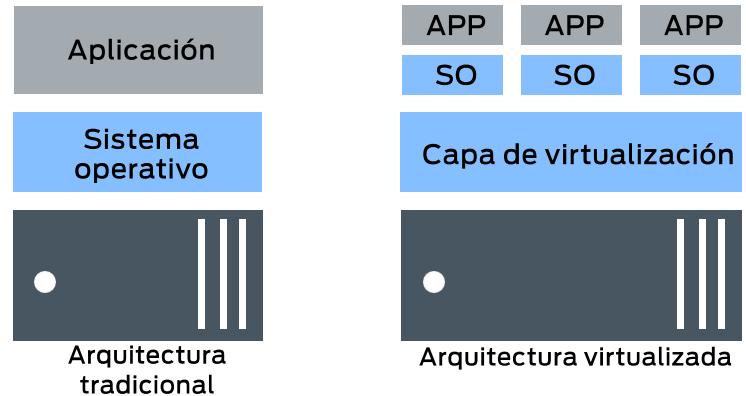


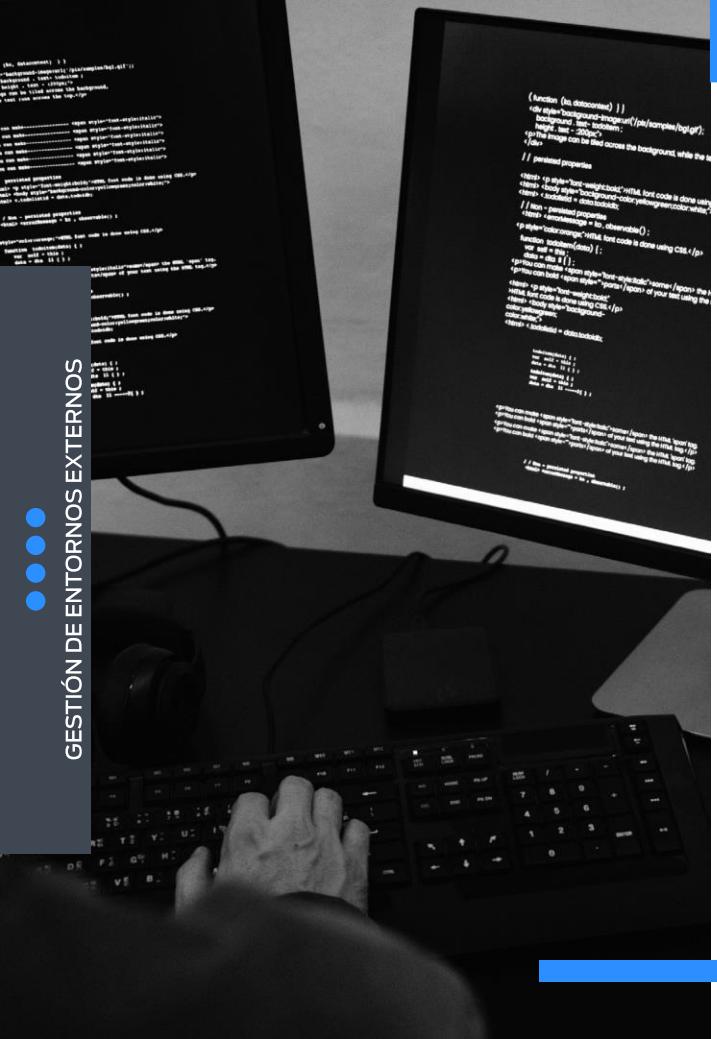
- El alumno será capaz de comprender de manera clara el funcionamiento básico de Amazon Linux.

- El alumno comprenderá el funcionamiento de la orquestación con DevOps.

Virtualización. Proceso en el cual se ejecuta una instancia virtual de un sistema informático en una capa que se abstracta del hardware. En este sentido, es la ejecución simultánea de múltiples sistemas operativos en un sistema informático. Así, las aplicaciones que se ejecutan en la parte superior de la máquina virtualizada parecerían estar en su propia máquina dedicada, en donde tanto el sistema operativo como las bibliotecas y otros programas son exclusivos del sistema virtualizado invitado, y no están conectados al sistema operativo host.

Arquitectura tradicional vs. arquitectura virtualizada



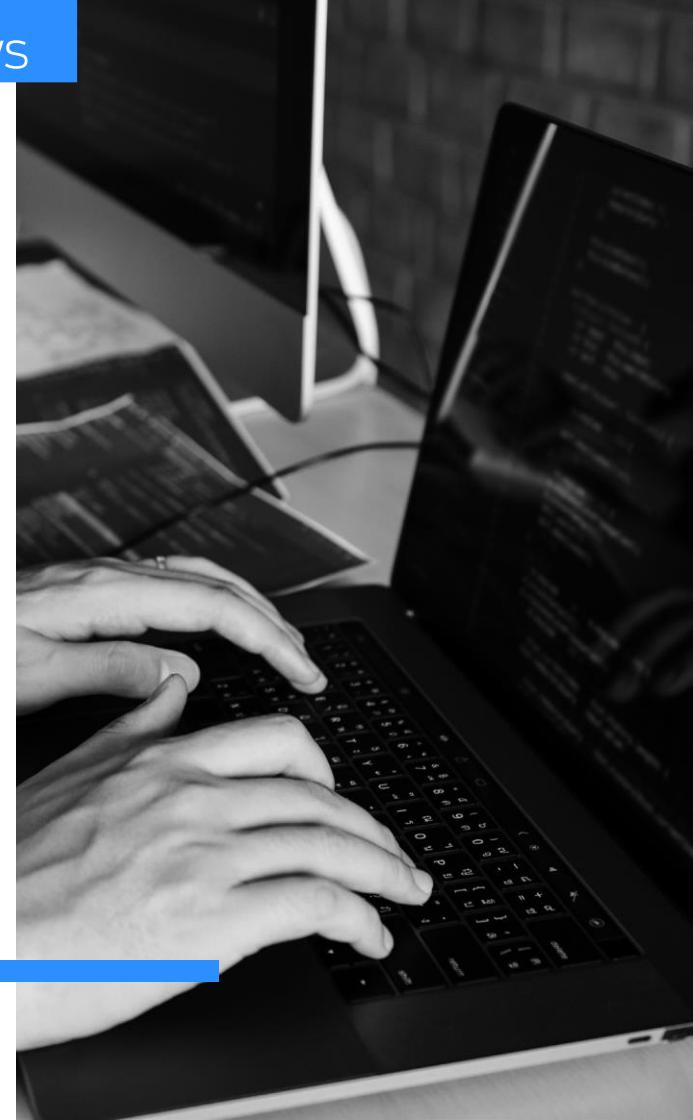


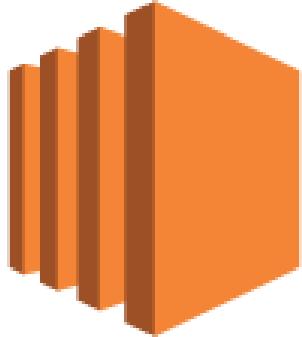
ALMACENAMIENTO EN AWS

Con base en lo anterior, la virtualización en la informática tienen múltiples usos. En general, uno de sus usos más comunes es poder ejecutar aplicaciones diseñadas para un sistema operativo diferente sin tener que cambiar de equipo o reiniciar en un sistema distinto.

Por su parte, para los **administradores de servidores** la virtualización también ofrece la capacidad de ejecutar diferentes sistemas operativos. No obstante, algo todavía más importante es la posibilidad de segmentar un sistema grande en muchas partes más pequeñas. Esto permite que una cantidad de usuarios diferentes utilice el servidor más eficientemente.

Aunado a ello, la virtualización permite el aislamiento, manteniendo los programas que se ejecutan dentro de una máquina virtual a salvo de los procesos que tienen lugar en otra máquina virtual en el mismo *host*.





Amazon
EC2

Amazon Elastic Compute Cloud (Amazon EC2):

- Es un servicio web que permite capacidad de cómputo de tamaño variable en la Nube.
- Es compatible con tecnologías de virtualización, por lo que puede usar una amplia variedad de sistemas operativos a través de sus interfaces de servicios web.
- Permite el uso escalable de aplicaciones, por lo que un usuario puede montar una imagen de máquina de Amazon para crear una máquina virtual (instancia), que contendrá cualquier software deseado.

Amazon Linux. Es una distribución respaldada y actualizada por *Amazon Web Services*. Está disponible para usar mediante las instancias de Amazon EC2.

AMI de Amazon Linux. Es una imagen de Linux compatible y mantenida, proporcionada por *Amazon Web Services* para su uso en Amazon EC2. Proporciona un entorno de tiempo de ejecución estable, seguro y de alto rendimiento.



Amazon Machine Image (AMI)

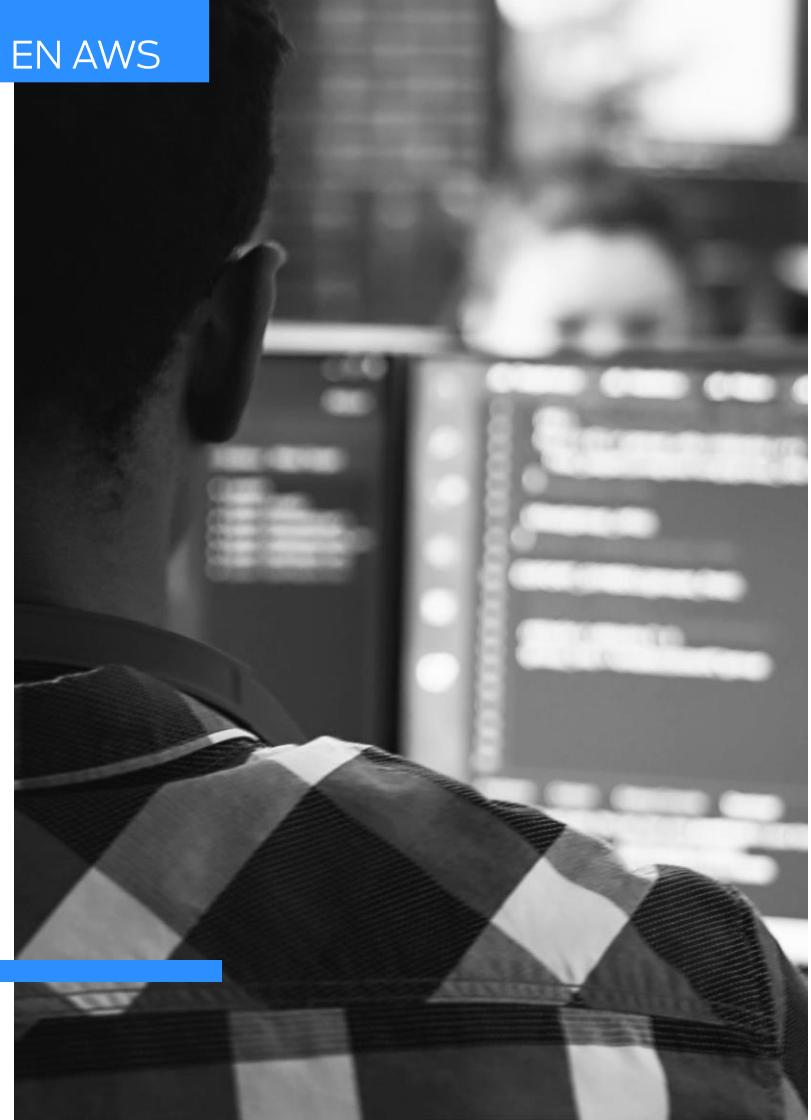


Características clave de la AMI de Amazon Linux:

- Incluye paquetes y configuraciones que brindan una estrecha integración con *Amazon Web Services*.
- Viene preinstalada con muchas herramientas API de AWS y CloudInit.
- Las herramientas de la API de AWS permiten la creación de secuencias de comandos de tareas de aprovisionamiento importantes desde una instancia de Amazon EC2.

La **configuración de la AMI de Amazon Linux** mejora la seguridad al centrarse en 2 objetivos de seguridad principales:

- 1. Limitar el acceso.** Mediante el uso de pares de claves SSH y la desactivación del inicio de sesión raíz remoto.
- 2. Reducir las vulnerabilidades del software.** Reduce la cantidad de paquetes no críticos que se instalan en su instancia.





ALMACENAMIENTO EN AWS

Aunado a lo anterior, la AMI de Amazon Linux incluye paquetes y configuraciones que brindan una integración perfecta con *Amazon Web Services*. Lo anterior permite que esta se lance y funcione con varios servicios de AWS listos para usar.

Los **repositorios**, por su parte, están disponibles en todas las regiones y se accede a ellos a través de **Yum**. Además, su alojamiento en cada región permite que las actualizaciones se implementen rápidamente y sin cargos por transferencia de datos.

Actualizaciones de seguridad:

- Se proporcionan a través de los repositorios Yum de la AMI de Amazon Linux.
- Las alertas de seguridad se publican en el centro de seguridad de la AMI de Amazon Linux.





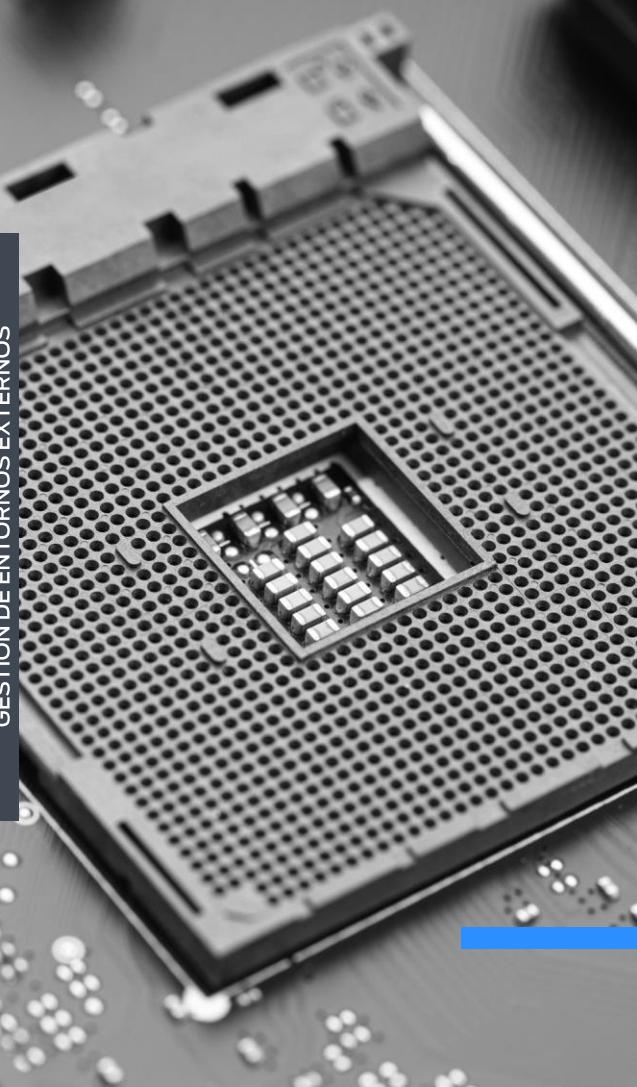
¿Para qué sirve Amazon EC2?:

1. Básicamente, este servicio web ofrece la posibilidad de crear entornos informáticos virtuales en la Nube.
2. Esto se logra a través de una interfaz web que se conecta a una imagen de máquina de Amazon AMI.
3. La AMI suele corresponder al sistema operativo que queremos ejecutar en la máquina virtual.

Existen **distintas instancias** en Amazon EC2, algunas de las cuales son:

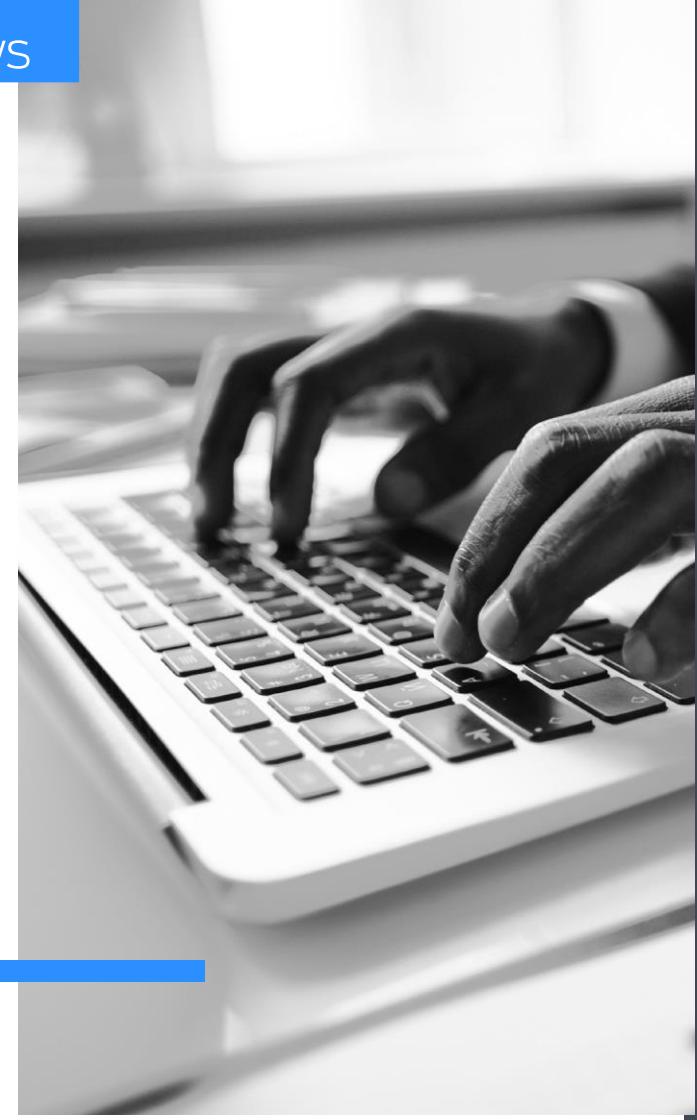
- Instancias T2
- Instancias M3 y M4
- Instancias C3 y C4
- Instancias R3
- Instancias G2
- Instancias I2 y D2





- **Instancias T2.** La CPU de estas instancias tiene un rendimiento fundamental, no obstante, es viable conseguir picos de actividad por arriba de lo regular. Se propone para aplicaciones y programas que no usen la CPU al 100%, pero que necesitan un rendimiento puntual máximo.
- **Instancias M3 y M4.** Ofrecen un rendimiento estable de alta capacidad. Útil para aplicaciones que requieren que la CPU y memoria permanezcan equilibradas en un ámbito de rendimiento exigente.

- **Instancias C3 y C4.** Unen la tecnología de los procesadores de alta frecuencia E5-2680 v2 y E5-2666 v3 de Intel para la utilización de aplicaciones que necesiten un rendimiento exhaustivo.
- **Instancias R3.** Útiles para aplicaciones que necesiten un rendimiento de memoria exhaustivo, como procesamiento analítico *in-memory*, bases de datos de elevado rendimiento o enormes instalaciones de aplicaciones.





ALMACENAMIENTO EN AWS

- **Instancias G2.** Están orientadas a aplicaciones que hagan un uso exhaustivo de la capacidad gráfica de la GPU.
- **Instancias I2 y D2.** Optimizadas para el almacenamiento de información, por ejemplo bases de datos.

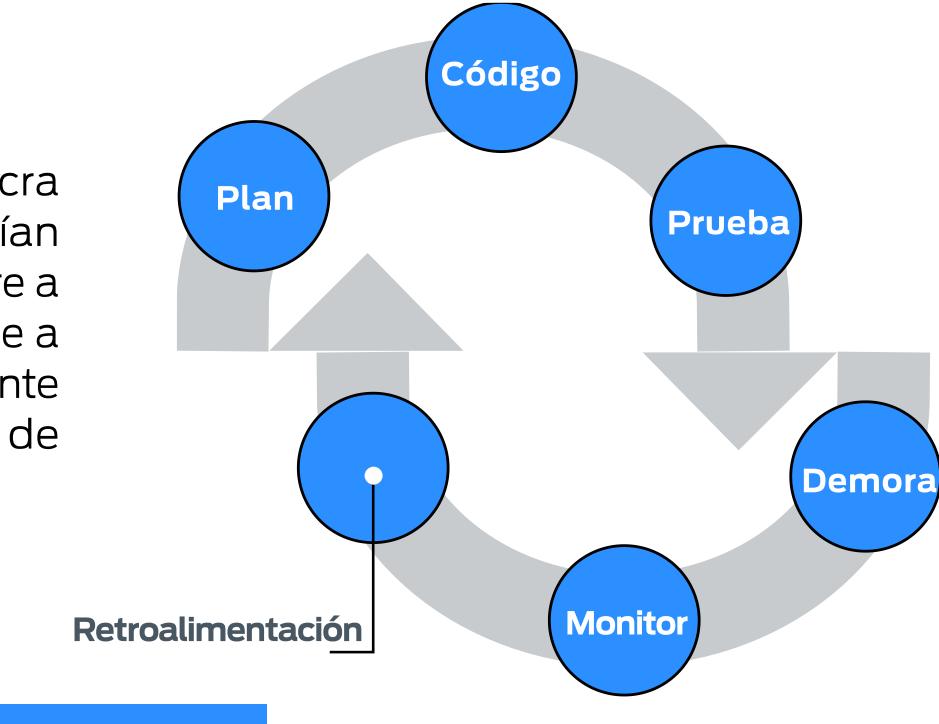


VIDEO

Te invitamos a ver el siguiente video:

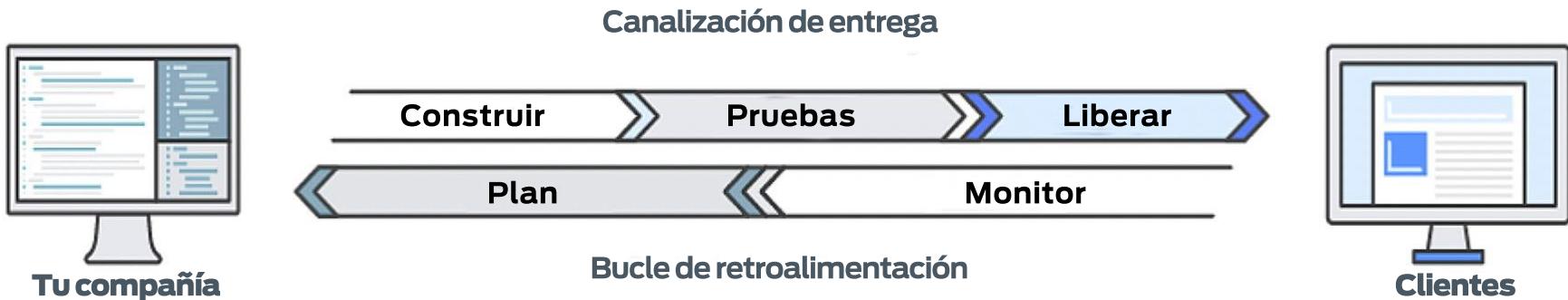


Para un proyecto de TI que involucra a varios desarrolladores que envían regularmente versiones de software a varios servidores existe un enfoque a adoptar: **DevOps**. Este es sumamente útil para la gestión de proyectos de software.

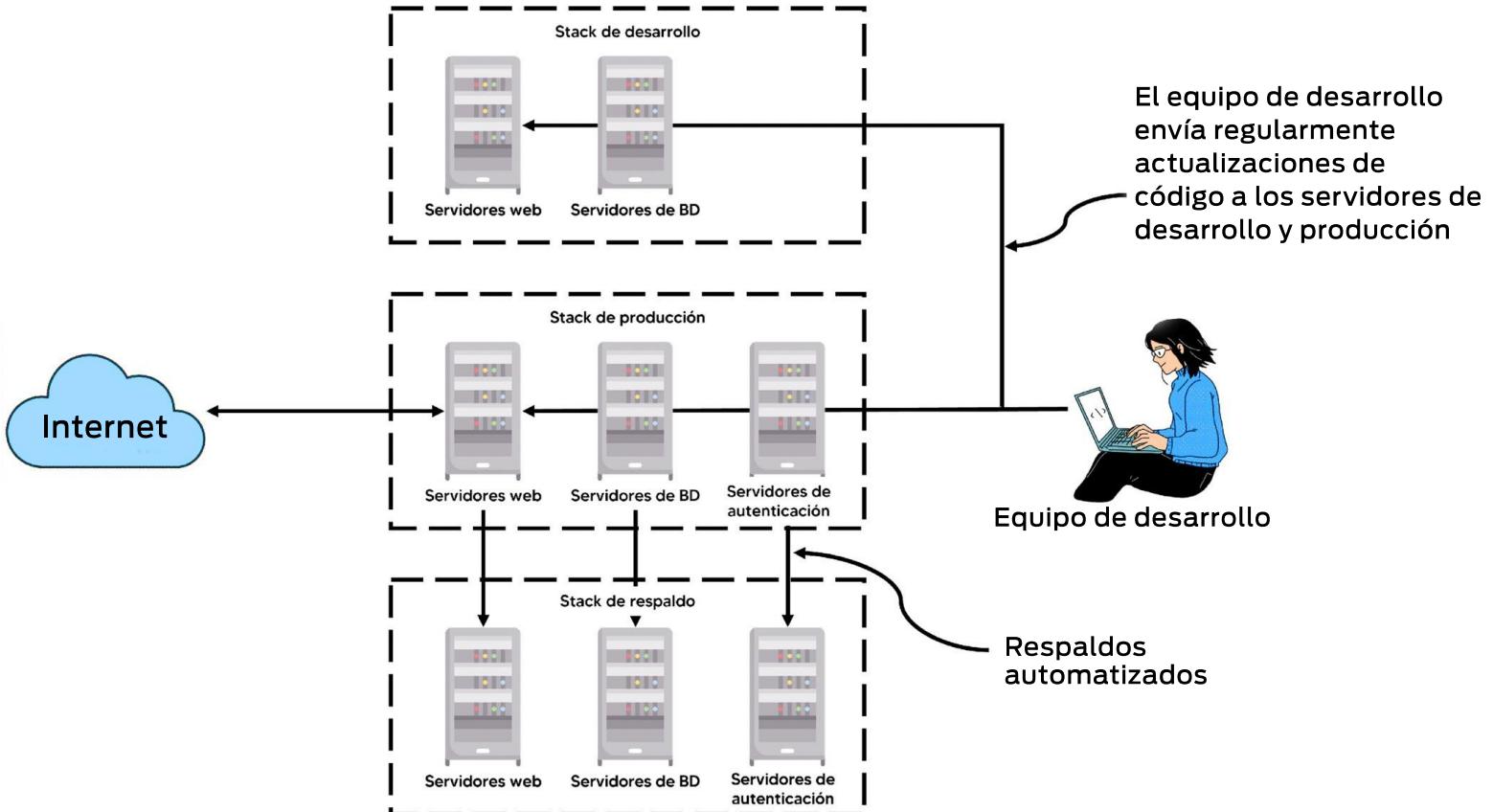


¿Qué es DevOps?:

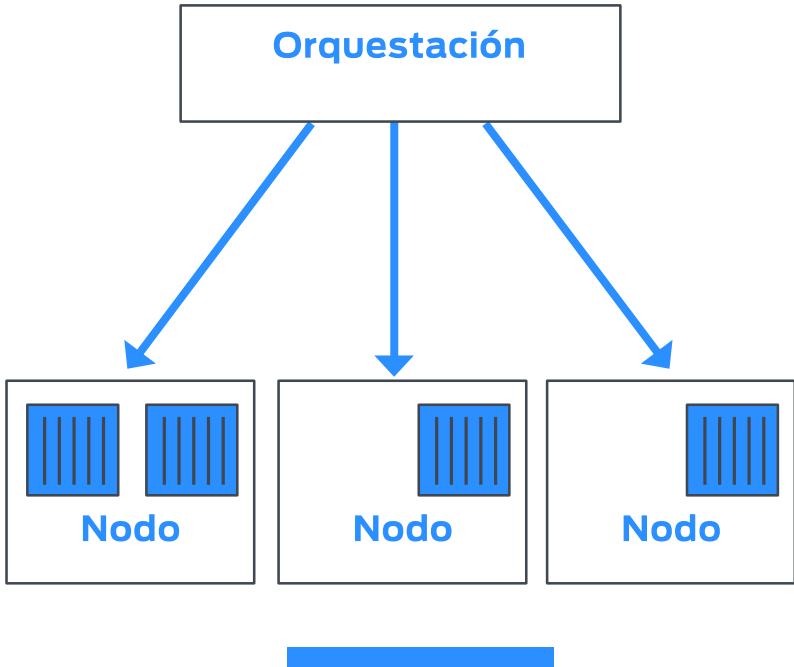
Es una combinación de filosofías, culturas, prácticas y herramientas que, en conjunto, aumentan la capacidad de una organización para entregar aplicaciones y servicios a alta velocidad. Permite evolucionar y mejorar los productos a un ritmo más rápido que las empresas tradicionales, que utilizan procesos tediosos de gestión de infraestructura y desarrollo de software. En consecuencia, el tiempo ahorrado permite brindar un mejor servicio a los clientes y competir de manera más efectiva en el mercado.



DESPLIEGUE DE APLICACIONES CON DEVOPS



Un entorno típico de desarrollo de aplicaciones alimentado por actualizaciones periódicas de software del equipo de desarrollo



Orquestación

No significa que vayamos a ponernos a escuchar alguna sinfonía o algo por el estilo, sino que, es el nombre que se le da al uso de las aplicaciones.

Imaginemos que necesitamos varios servicios, por lo que, gracias a la orquestación, podemos usar varias máquinas físicas; una hace de servidor *proxy*; otra de base de datos; otra de *backend*.

Eso es orquestación, y se puede hacer de varias formas y con varios sistemas.

DevOps parte de la idea de quitar las separaciones entre Dev (*development*) y Ops (*operations*). Para lograrlo, se propone el análisis **QA** (*Quality Assurance*). Este último viene acompañado con métricas y pruebas; además, tiene un alto costo de recursos.

En este sentido, para que la solución sea posible, la única forma es **automatizar los procesos**. Es en este contexto que surge el concepto de **orquestación**.

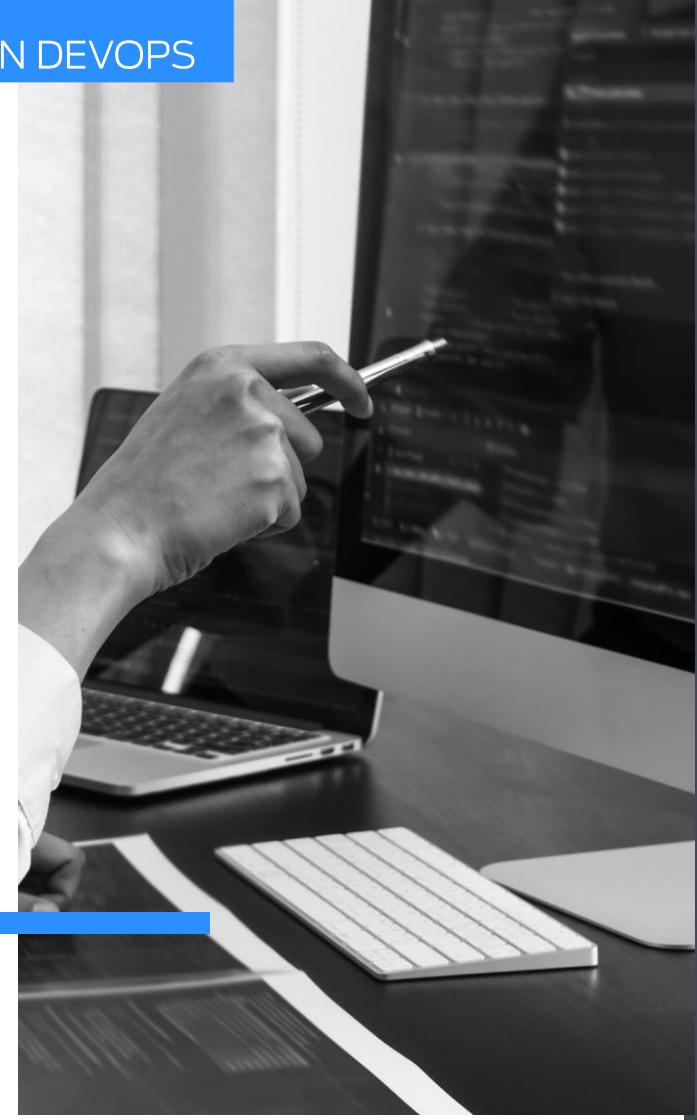




¿Por qué Linux es tan importante en DevOps?:

- Linux se usa en casi todo.
- La infraestructura de las empresas es responsabilidad de los ingenieros DevOps.
- Por ser un sistema operativo de *open source*, es altamente maleable y flexible.
- Es altamente escalable.
- Es compatible con las principales herramientas DevOps.

- **Linux se usa casi en todo.** Ya sea un desarrollador internacional de software o un consumidor de aplicaciones para teléfonos inteligentes, probablemente esté usando Linux. En la actualidad, este sistema operativo está detrás de muchas de las tecnologías que están en el corazón de dispositivos y servicios, desde *smartphones*, aplicaciones de los principales proveedores de servicios de contenido, herramientas de productividad personal, redes sociales, etcétera.



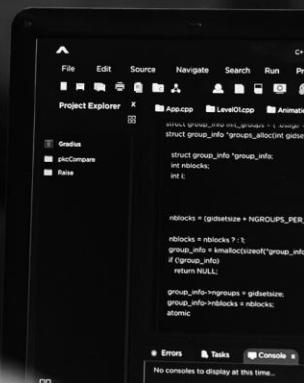


- **La infraestructura de las empresas es responsabilidad de los ingenieros DevOps.** En un departamento de TI no es posible elegir qué sistema operativo de la infraestructura se va a usar a la hora de realizar un despliegue o atender una incidencia en una de las aplicaciones que tiene la particularidad de estar alojada en un servidor Linux.



- **Por ser un sistema operativo de *open source*, es altamente maleable y flexible.** Dada la abertura de código y versatilidad que tiene Linux, se posibilita que pueda ser modificado a placer, dando la libertad de poder cambiarse de aplicación o plataforma. A partir del diseño de un flujo de trabajo en especial, las aplicaciones que ocupe servir, o protocolos de estabilidad específicos que se deseen llevar a cabo.



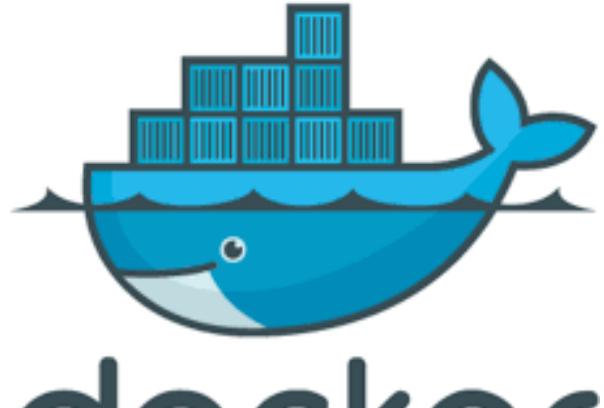
A screenshot of a terminal window with a dark background. The window title is 'Code'. The terminal shows a C code snippet for the Linux kernel. The code includes declarations for 'group_info' and 'group_info' structures, and a function that calculates the number of blocks based on group size and the number of groups per block. The code is annotated with comments explaining the logic. The bottom of the terminal shows tabs for 'Errors', 'Tasks', and 'Console', with a message 'No consoles to display at this time'.

- **Es altamente escalable.** El *Kernel* de Linux puede guardar y procesar gigantes porciones de información en memoria, y puede combinar numerosas tecnologías de almacenamiento de diferentes tecnologías en una sola unidad lógica de almacenamiento. La escalabilidad es fundamental para que la CI/CD (o Integración Continua / Entrega Continua) puedan ser llevadas a la práctica en la operación DevOps. Afortunadamente, Linux es un sistema operativo escalable.

- **Es compatible con las principales herramientas DevOps.** Algunos ejemplos de las más utilizadas son:

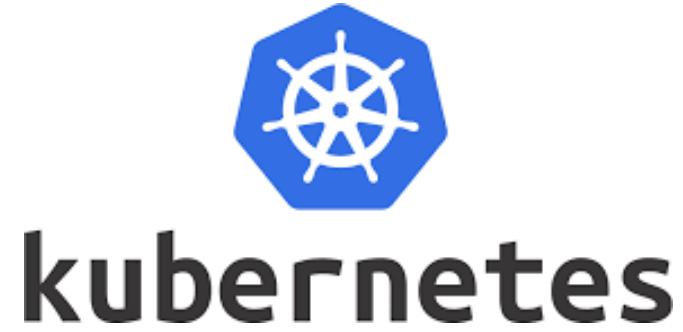
- Docker
- Kubernetes
- Ansible
- Puppet





- **Docker.** Esta plataforma de contenedores de aplicaciones es compatible con Linux en un gran nivel. Incluso varios comandos que se aplican para echar a andar contenedores, tirar de imágenes para proporcionar redundancia de servicios y otro tipo de configuraciones poseen mucha afinidad con ciertos comandos de Linux. Por consiguiente, su sintaxis se aprende más rápidamente por un cliente de Linux.

- **Kubernetes.** Este entorno de trabajo es muy útil para realizar la orquestación de contenedores, ya que usa Linux de forma casi nativa. Sin embargo, para lograr llevar a cabo dichos mismos ambientes de trabajo en otras plataformas hacen falta más configuraciones y preparación extras.





ANSIBLE



- **Ansible y Puppet.** Ambas plataformas se aplican para hacer configuraciones u operaciones con una cantidad enorme de *servers* en paralelo por medio de IaaS. Piden en sus requerimientos que las máquinas primordiales encargadas de realizar los despliegues de configuraciones masivas sean servidores Linux exclusivamente, ningún otro sistema operativo es aceptado.

LECTURAS PARA REFORZAR LA UNIDAD



Capítulo 1

- *Amazon Web Services.* (2020, 9 septiembre). Amazon Linux. Amazon Web Services, Inc.
- *IBM.* (2021, 3 septiembre). *Amazon AWS EC2.* © Copyright IBM Corp. 2013, 2015.



LECTURAS PARA REFORZAR LA UNIDAD



Capítulo 2

- Pathak, A. (2022, 11 abril). *Explora las 30 Mejores Herramientas de DevOps a Tener en Cuenta en 2022*. Kinsta.
- Amazon Web Services. (2020, 26 octubre). *What is DevOps?*





CONCLUSIÓN

En las últimas fechas, Amazon se encuentra cada vez más dentro del mundo del desarrollo de aplicaciones, debido a sus instalaciones rápidas que permiten a los usuarios acceder a la infraestructura informática en cuestión de minutos a bajo costo. También permite a los clientes aumentar y disminuir fácilmente la capacidad durante un largo período de tiempo, y tiene una gama de herramientas para asegurar la privacidad de los datos.

Por otro lado, con Linux, DevOps obtiene la escalabilidad y flexibilidad que necesita para tener un proceso de desarrollo de software dinámico. El sistema operativo permite configurar el entorno de desarrollo para satisfacer sus necesidades. En lugar de permitir que el sistema operativo dicte cómo funcionan los equipos de DevOps, puede configurar Linux para que funcione para ellos.



¡FELICIDADES!

Acabas de concluir la **cuarta unidad** de tu curso *Sistemas Operativos II*. Te invitamos a finalizar este esfuerzo realizando el examen parcial correspondiente. Para ello, debes regresar a la pantalla principal y dar clic en *Presentar examen*.

BIBLIOGRAFÍA



- Amazon. (2019, 26 febrero). AWS | Amazon Linux AMI. Amazon Web Services, Inc.
- Open Source. (2018, 25 octubre). *What is virtualization?* Opensource.Com
- Clinton, D. (2018). *Linux in Action* (Illustrated ed.). Manning Publications.
- Escobar, R. A. S. (2022, 1 febrero). *La importancia de Linux para la ingeniería DevOps*. OpenWebinars.net
- Katalon Team. (2021, 15 junio). *DevOps Orchestration: How To Improve Your Automation*. Katalon.

BIBLIOGRAFÍA



- Saive, R. (2021, 6 agosto). 20 *Command Line Tools to Monitor Linux Performance*. TechM ind.
- G. (2022a, enero 24). *Solucionar problemas de rendimiento y aislar cuellos de botella en Linux - Virtual Machines*. Microsoft Docs.
- NIAZI, R. U. M. A. I. S. A. (2022b, febrero 21). *The 9 Best Linux Network Troubleshooting Commands*. Make Use Of.



BIBLIOGRAFÍA



- Vivek Ghate, S. (2011). *Operating System Concepts and Basic Linux Commands*. EDUCREATION PUBLISHING.
- Nordhoff, A. (2020, 13 julio). *What is a Cluster? An Introduction to Clustering in the Cloud*. Capital One.
- Dietz, H. (2019, 14 febrero). *Linux Parallel Processing HOWTO: Introduction*. The Linux Documentation Project.



BIBLIOGRAFÍA

- Silberschatz, A., Baer Galvin, P., & Gagne, G. (2013). *Operating System Concepts*. (Ninth Edition). Wiley.
- Argüello, F. (2022, 8 enero). *Guía de Seguridad Servidores Linux*. Infoteknico.

PROYECTO FINAL

A man in a dark suit and tie is using a tablet computer. He is pointing with his right index finger. The background is blurred, showing an office environment with a computer monitor. A solid blue horizontal bar is positioned across the middle of the image, containing the text "PROYECTO FINAL".



Te invitamos a realizar el proyecto final:

Presiona el botón para descargar el proyecto final:



Presiona el botón para entregar el proyecto final:

