

# Diferencias entre Windows y Linux

**DigitalHouse** >  
Coding School



**Certified Tech  
Developer**  
The Ultimate Degree

# Índice

1. [Diferencias y no tantas](#)
2. [Interfaz de usuario](#)
3. [Filosofía y licenciamiento](#)
4. [Sistema de paquetes](#)
5. [Aspectos técnicos](#)

1

# Diferencias y no tantas

## “Linux es un cáncer”

Steve Ballmer, CEO de Microsoft , Junio de 2001.

Tomando como posta la frase de Steve Ballmer, y ubicándonos en un contexto histórico, las diferencias entre los dos sistemas operativos más utilizados en el mundo de los servidores siempre existieron. Y pasaron desde cosas sencillas como nombres de comandos a cuestiones técnicas mucho más profundas.

Y a lo largo de esta presentación, veremos cómo algunas de ellas increíblemente fueron desapareciendo, alejándose de esa frase —que hacía referencia a cuestiones de paquetes de Linux—.

2

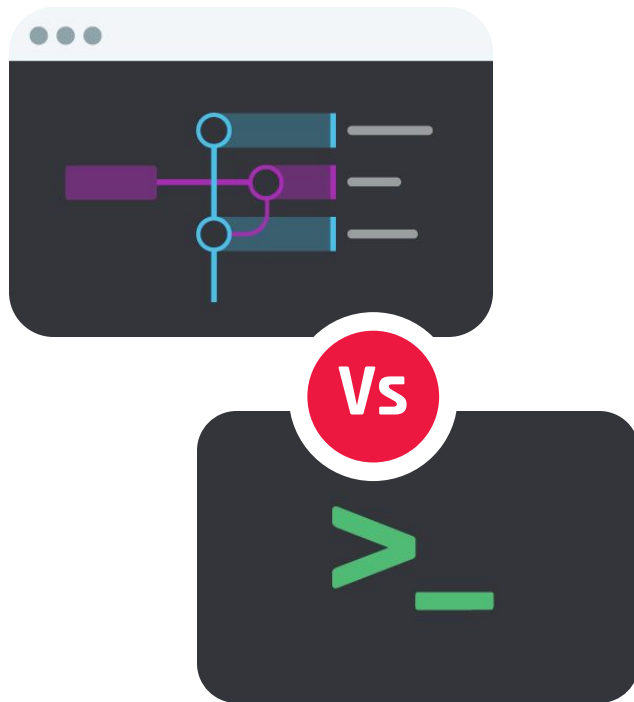
## Interfaz de usuario

# Interfaz de usuario

Históricamente, la primer gran diferencia fue la interfaz de usuario, la barrera más grande, la que durante mucho tiempo hizo que pareciera que aquellos que administraban Linux eran cuasi “hackers” y los de Windows, newbies.

Siempre la relación era: Windows igual a interfaz gráfica y Linux igual a consola.

Lo cual no solo diferenciaba las maneras de interactuar, sino que formaba a perfiles de administradores muy diferentes.



# Linux: Shell

Como heredero de UNIX, Linux posee una interfaz de línea de comandos, denominada Shell. Super potente, la cual es la puerta de entrada a la administración del sistema.

Soporta nativamente un componente de script, llamado Bash, el cual permite optimizar y automatizar multiplicidad de trabajos tales como mantenimientos o recuperaciones ante caída de servicios.

Este Shell no solo se inicia por defecto en cualquier entorno de servidores, sino que es accesible por métodos tan arcaicos como una comunicación por puerto serial.

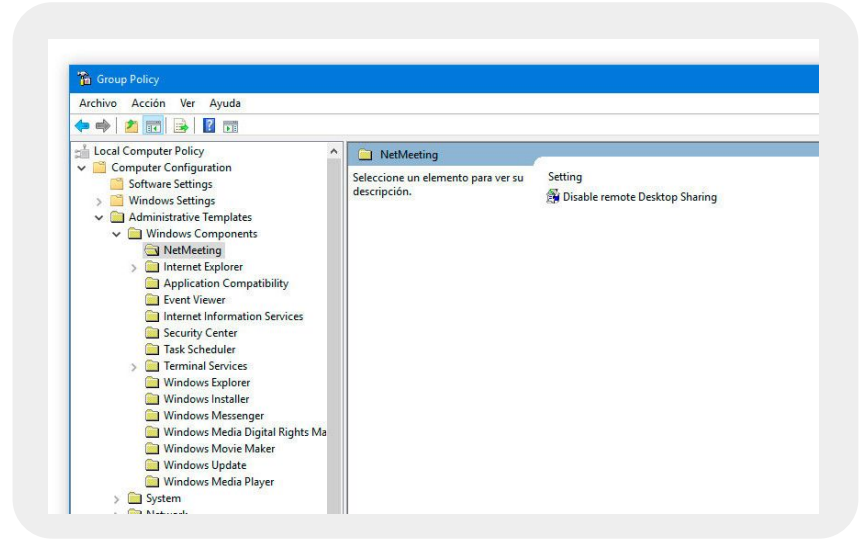
[illegible]

# Windows: GUI (interfaz gráfica)

Windows históricamente fue un sistema operativo que se administró con una interfaz gráfica completa, la cual se basó en consolas que tenían funciones específicas.

Esta manera de administrar Windows Server, muy similar al Windows “doméstico”, hace que la curva de aprendizaje sea mucho más suave, sin necesidad de aprender difíciles comandos.

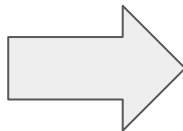
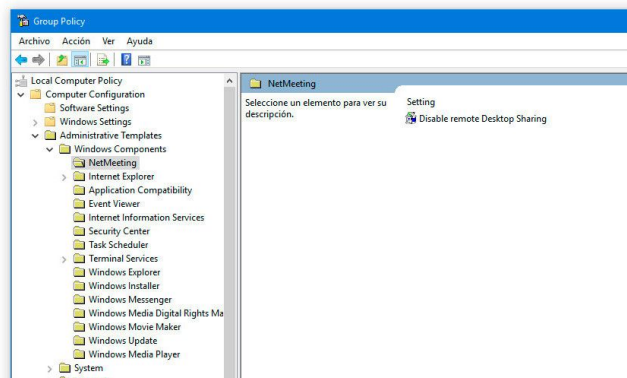
La interfaz gráfica tuvo poca evolución. Recién en versiones más recientes se han migrado algunas administraciones hacia una nueva interfaz, en línea a las “app settings”.





# ¿Se acercaron en algo?

En la actualidad, podemos decir que el acercamiento más grande lo ha hecho Windows, ya que incorpora una interfaz de consola muy potente, llamada **Powershell**, la cual permite la administración completa del servidor, dejando a la antigua CMD como algo obsoleto.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> get-help get-service

NAME
Get-Service

SYNOPSIS
Gets the services on a local or remote computer.

SYNTAX
Get-Service [-ComputerName <string>] [-DependentServices] -DisplayName <string> [-Exclude <string>]
[-Include <string>] [-RequiredServices] [-CommonParameters]

Get-Service [-ComputerName <string>] [-DependentServices] [-Exclude <string>] [-Include <string>]
[-InputObject <servicecontroller>] [-RequiredServices] [-CommonParameters]

Get-Service [-Name <string>] [-ComputerName <string>] [-DependentServices] [-Exclude <string>] [-Include
<string>] [-RequiredServices] [-CommonParameters]

DESCRIPTION
The Get-Service cmdlet gets objects that represent the services on a local computer or on a remote computer,
including running and stopped services.

You can direct this cmdlet to get only particular services by specifying the service name or display name of the
services, or you can pipe service objects to this cmdlet.

RELATED LINKS
Online Version: http://go.microsoft.com/fwlink/?LinkID=821593
New-Service
Restart-Service
Resume-Service
Set-Service
Start-Service
Stop-Service
Suspend-Service

REMARKS
To see the examples, type: "get-help Get-Service -examples".
For more information, type: "get-help Get-Service -detailed".
For technical information, type: "get-help Get-Service -full".
For online help, type: "get-help get-service -online".

PS C:\Windows\system32>
```

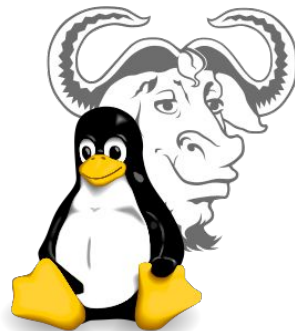
3

## Filosofía y licenciamiento

# Linux: Open source, mayormente gratuito

Linux es un software de cuya filosofía es de código abierto (open source), esto quiere decir que dicho código está a disposición para ser revisado, modificado y mejorado. En el único caso que debemos pagar es para obtener soporte técnico específico.


Su denominación técnica (GNU/Linux) deriva de los proyectos GNU (sistemas open source derivados de UNIX) y del kernel Linux.



# Linux: Muchos derivados, misma esencia

Partiendo de una misma base, aparecen decenas de distribuciones GNU/Linux, las cuales se originan en un proceso en donde se toma el núcleo Linux y se le agregan determinados paquetes de interés.

Esto permite que existan tantas distribuciones como necesidades, e inclusive, ya existen empaquetados que al instalarse no solo incluyen el sistema operativo base, sino que agregan funcionalidades extras por defecto.

<b>Debian</b>	
<b>Ubuntu</b>	 ubuntu
<b>RedHat</b>	 Red Hat
<b>SUSE</b>	

# WINDOWS: Privativo, propietarios y pago

Desde sus inicios, Windows Server fue un sistema operativo de código cerrado (nadie más que el mismo Microsoft puede modificarlo), además de tener un esquema de licenciamiento arancelado, el cual se basa no solo por equipo, sino por núcleo de proceso.

El pago de esta licencia permite tener soporte sobre el sistema operativo, lo cual lo hace preferido en ambientes corporativos.



# Windows: Un solo gusto, varias intensidades

En su base, todos los Windows Server (considerando la última versión lanzada), son iguales, pero, en pos de ofrecer una variedad de funcionalidades y costos, Microsoft comercializa diferentes tipos de licencia, las cuales permiten acceder a ciertas funciones del sistema o poder aprovechar todos los núcleos de proceso de un servidor. Veamos algunos ejemplos:

<b>Essentials</b>	Apuntado para empresas de hasta 25 clientes.
<b>Standard</b>	Sin límite de clientes, pero con funciones de virtualización limitadas.
<b>Datacenter</b>	La más completa, destaca el soporte completo para virtualización y contenedores.

# 4

## Sistema de paquetes

# LINUX: Gestión de paquetes, repositorios

Una de las características más destacadas en Linux es la gestión de paquetes de software, la cual permite de manera muy precisa que se instale la versión/compilación de software más adecuada a nuestra distribución.

Tiene dos componentes principales

- **Gestor de paquetes:** es quien se encarga de administrar las dependencias (por ejemplo, si tal o cual software precisa otro paquete adicional) y comparar nuestras versiones instaladas con las últimas publicadas.
- **Repositorios:** permiten ofrecer un catálogo de aplicaciones gestionado desde un lugar centralizado y supervisado por la comunidad de desarrolladores y usuarios. Esto da garantías al usuario de que todo el software que ha descargado es legítimo, y que además estará soportado.



# LINUX: 3 grandes jugadores

A pesar de haber decenas de distribuciones “importantes”, en lo que se refiere a tipos de paquetes, existen 3:

## **DPKG (Debian Package Manager):**

originalmente creado para Debian, y utilizado por sus derivados como Ubuntu, Linux Mint o Elementary Os.

## **RPM (Red Hat Package Manager):**

creado por RedHat, utilizado en sus propias distribuciones, como así también en Fedora y OpenSuse.

## **Pacman Package Manager:**

creado para ArchLinux, y presente en la mayoría de las distribuciones “livianas”, ya que aprovecha muy bien los recursos.

# WINDOWS: Un poco más centralizado



Uno de los beneficios de ser una plataforma más cerrada es la estandarización de paquetes, en este caso se centran en archivos ejecutables (extensión .exe) o paquetes msi de Windows Installer (extensión .msi)

Si bien con el paso del tiempo y la acumulación de versiones, suele diversificarse las versiones, en general existen estos paquetes para un software Windows:

Arquitectura	Compatible con
32 bit	Windows 2000 y 2003
64 bit	Windows 2003 64 bit, 2008, 2012, 2016 y 2019

# WINDOWS: Gestores de paquetes alternativos

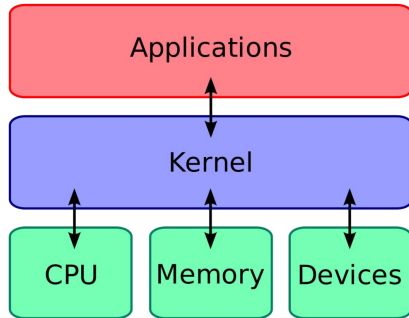
Con el paso del tiempo, aparecieron gestores de paquetes desarrollados por terceros, que emulan el comportamiento de la gestión de paquetes de Linux, en pos de facilitar el mantenimiento y la actualización del software instalado. Los más destacados son:

Chocolatey	Ninite
 <p>Escrito en .NET Framework, utiliza la infraestructura de empaquetado de NuGet y Windows PowerShell para proporcionar una herramienta de automatización e instalar software en máquinas Windows y fue diseñado para simplificar el proceso desde la perspectiva del usuario. Es gratuito y se administra principalmente por línea de comandos.</p>	 <p>Es un empaquetador de instaladores, cuyo propósito principal es que las instalaciones sean los más desatendidas posibles. Su funcionamiento se basa seleccionar qué paquetes queremos instalar en la propia web de Ninite y se generará un megainstalador. Tiene una versión gratuita y una paga, que está apuntada hacia el despliegue de estos paquetes de manera masiva en varios ordenadores.</p>

# 5 | Aspectos técnicos

# Diferencias técnicas

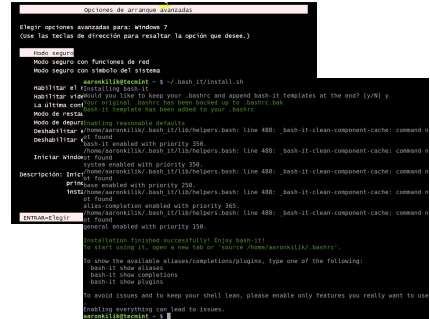
Las diferencias técnicas se agrupan en 3 aspectos más distintivos de cada sistema operativo:



# Kernel



# Sistemas de archivos



## Niveles de ejecución

# Kernel

Es el software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado. Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora.

Windows	Linux
<ul style="list-style-type: none"><li>• En dos capas, una de kernel y otra de usuario.</li><li>• Usa una lista de control de acceso.</li><li>• Incluye el kernel los aspectos relativos a la interfaz gráfica.</li><li>• Almacena sus configuraciones en registros.</li></ul>	<ul style="list-style-type: none"><li>• De una sola capa, la del propio kernel.</li><li>• Usa los permisos tradicionales de Unix para el control de acceso a archivos.</li><li>• Los aspectos relativos a la interfaz gráfica se apoya en la capa de usuario.</li><li>• Almacena sus configuraciones en archivos.</li></ul>

# Sistema de archivos: el punto al almacenamiento

Es el componente del sistema operativo encargado de administrar y facilitar el uso de las memorias periféricas.

Además estructura y organiza la escritura, búsqueda, lectura, almacenamiento, edición y eliminación de archivos de una manera concreta. El objetivo principal de esta organización es que el usuario pueda identificar los archivos sin lugar a error y acceder a ellos lo más rápido posible.

Windows	Linux
<ul style="list-style-type: none"><li>• <b>FAT/ FAT16 / FAT32</b>, heredado desde el MS-DOS. Tiene un límite máximo de 4 GB por archivo.</li><li>• <b>NTFS</b>, el más utilizado en Windows ya que brinda la posibilidad de tener campos específicos para incorporar permisos de Active Directory, en la actualidad va por su 3er evolución, incorporando soporte nativo para SSD.</li></ul>	<ul style="list-style-type: none"><li>• <b>EXT</b>, el cual esta ya es su cuarta evolución, es el más utilizado ya que tiene amplio soporte para las unidades SSD.</li><li>• <b>ReiserFS</b>, de propósito general, apunta a la baja corrupción de archivos.</li><li>• <b>swap</b>, es el sistema de archivos para los archivos de intercambio.</li></ul>

# Niveles de ejecución: controlando al S.O.

El nivel de ejecución de un sistema (también conocido como estado init) define qué servicios y recursos están disponibles para los usuarios. Un sistema solo puede estar en un nivel de ejecución a la vez. Si en Windows se denominan niveles de inicio, cumplen la misma función que los de ejecución.

Windows	Linux
<ul style="list-style-type: none"><li>• <b>Modo seguro con funciones de red.</b> Inicia en modo seguro e incluye los controladores y servicios de red</li><li>• <b>Modo seguro con símbolo del sistema.</b> Inicia Windows en modo seguro con una ventana de símbolo del sistema en lugar</li><li>• <b>Habilitar el registro de arranque.</b> Crea un archivo llamado ntbtlog.txt en el que puede resultar útil para la solución avanzada de problemas.</li><li>• <b>La última configuración válida conocida.</b> Inicia Windows con la última configuración del registro y los controladores que funcionó correctamente.</li><li>• <b>Deshabilitar reinicio automático tras error.</b> Impide que se reinicie automáticamente en caso de que un error</li><li>• <b>Iniciar Windows normalmente.</b></li></ul>	<ul style="list-style-type: none"><li>• <b>Nivel 0:</b> Sistema Apagado</li><li>• <b>Nivel 1:</b> Solo usuario root, sin red ni demonios</li><li>• <b>Nivel 2:</b> Multiusuario sin red</li><li>• <b>Nivel 3:</b> Inicio normal, con consola</li><li>• <b>Nivel 5:</b> Inicio normal, con consola y además interfaz gráfica</li><li>• <b>Nivel 6:</b> Reinicio del sistema</li></ul>



DigitalHouse>  
Coding School