

Ejecución de los comandos para visualizar servicios en Windows:

Servicios Activos por defecto:

```
PS C:\WINDOWS\system32> Get-WmiObject Win32_Service | Where-Object { $_.StartMode -eq "Auto" } | Select-Object Name, State, StartMode
```

Name	State	StartMode
AMD Crash Defender Service	Running	Auto
AMD External Events Utility	Running	Auto
AudioEndpointBuilder	Running	Auto
Audiosrv	Running	Auto
BFE	Running	Auto
BITS	Running	Auto
BrokerInfrastructure	Running	Auto
CDPSvc	Running	Auto
CoreMessagingRegistrar	Running	Auto
CryptSvc	Running	Auto
DcomLaunch	Running	Auto
Dhcp	Running	Auto
DiagTrack	Running	Auto
DispBrokerDesktopSvc	Running	Auto
Dnscache	Running	Auto
DoSvc	Running	Auto
DPS	Running	Auto
DusmSvc	Running	Auto
edgeupdate	Stopped	Auto
EventLog	Running	Auto
EventSystem	Running	Auto
FontCache	Running	Auto
GoogleUpdaterInternalService138.0.7194.0	Stopped	Auto
GoogleUpdaterService138.0.7194.0	Stopped	Auto
gpsvc	Running	Auto
InventorySvc	Running	Auto
iphlpvc	Running	Auto
LanmanServer	Running	Auto
LanmanWorkstation	Running	Auto
LocalKdc	Stopped	Auto
LSM	Running	Auto
MapsBroker	Stopped	Auto
MDCoreSvc	Running	Auto
mpssvc	Running	Auto
msi	Running	Auto
PcaSvc	Running	Auto
Power	Running	Auto
ProfSvc	Running	Auto
RpcEptMapper	Running	Auto
RpcSs	Running	Auto
SamSs	Running	Auto
Schedule	Running	Auto
SENS	Running	Auto
ShellHWDetection	Running	Auto
Spooler	Running	Auto
sppsvc	Stopped	Auto
StateRepository	Running	Auto
StorSvc	Running	Auto
SysMain	Running	Auto
SystemEventsBroker	Running	Auto
TextInputManagementService	Running	Auto
Themes	Running	Auto
TrkWks	Running	Auto
UserManager	Running	Auto
UsoSvc	Running	Auto
Wallpaper Engine Service	Running	Auto
Wcmsvc	Running	Auto
WinDefend	Running	Auto
Winmgmt	Running	Auto

```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> Get-WmiObject Win32_Service | Where-Object { $_.Name -eq "wuauserv" } | Select-Object *

PSComputerName      : DESKTOP-ET73PPA
Name                 : wuauserv
Status              : OK
ExitCode             : 0
DesktopInteract      : False
ErrorControl         : Normal
PathName             : C:\WINDOWS\system32\svchost.exe -k netsvc -p
ServiceType          : Share Process
StartMode            : Manual
GENUS                : 2
CLASS               : Win32_Service
SUPERCLASS           : Win32_BaseService
DYNASTY              : CIM_ManagedSystemElement
RELPATH              : Win32_Service.Name="wuauserv"
PROPERTY_COUNT       : 26
DEVIATION            : 0
SERVER              : DESKTOP-ET73PPA
NAMESPACE            : root\cimv2
PATH                 : \\DESKTOP-ET73PPA\root\cimv2:Win32_Service.Name="wuauserv"
AcceptPause          : False
AcceptStop           : True
Caption              : Windows Update
CheckPoint           : 0
CreationClassName    : Win32_Service
DelayedAutoStart     : False
Description          : Habilita la detección, descarga e instalación de actualizaciones de Windows y otros programas. Si se deshabilita este servicio, los usuarios de este equipo no podrán usar Windows Update ni su
                        función de actualización automática y los programas no podrán usar la API del Agente de Windows Update (WUA).
DisplayName          :
InstallDate          :
ProcessId            : 11644
ServiceSpecificExitCode : 0
Started              : True
StartName             : LocalSystem
State                : Running
SystemCreationClassName : Win32_ComputerSystem
SystemName           : DESKTOP-ET73PPA
Tagid                : 0
WaitHint             : 0
Scope                : System.Management.ManagementScope
Path                 : \\DESKTOP-ET73PPA\root\cimv2:Win32_Service.Name="wuauserv"
Options              : System.Management.ObjectGetOptions
ClassPath            : \\DESKTOP-ET73PPA\root\cimv2:Win32_Service
Properties           : {AcceptPause, AcceptStop, Caption, CheckPoint...}
SystemProperties      : {__GENUS, __CLASS, __SUPERCLASS, __DYNASTY...}
Qualifiers            : {dynamic, local, provider, UUID}
Bits                 :
Container            :
```

Información detallada del proceso wuauserv.

Ejecución en Linux

```
31 de may 23:58

~ (1.314s)
systemctl list-unit-files --type=service | grep enabled
accounts-daemon.service      enabled          enabled
alsa-utils.service           masked          enabled
anacron.service              enabled          enabled
apparmor.service             enabled          enabled
apport.service               enabled          enabled
avahi-daemon.service         enabled          enabled
bluetooth.service            enabled          enabled
brltty.service               disabled         enabled
cloud-config.service          enabled          enabled
cloud-final.service           enabled          enabled
cloud-init-local.service      enabled          enabled
cloud-init.service            enabled          enabled
console-setup.service         enabled          enabled
cron.service                 enabled          enabled
cryptdisks-early.service     masked          enabled
cryptdisks.service           masked          enabled
cups-browsed.service          enabled          enabled
cups.service                 enabled          enabled
dmesg.service                enabled          enabled
e2scrub_reap.service         enabled          enabled
fancontrol.service           enabled          enabled
getty@.service               enabled          enabled
gnome-remote-desktop.service enabled          enabled
gpu-manager.service          enabled          enabled
grub-common.service           enabled          enabled
grub-initrd-fallback.service enabled          enabled
hwclock.service              masked          enabled
kerneloops.service           enabled          enabled
keyboard-setup.service        enabled          enabled
lm-sensors.service           enabled          enabled
lxc-monitor.service          enabled          enabled
lxc-net.service              enabled          enabled
lxc.service                  enabled          disabled
lxc@.service                  disabled         enabled
lxcfs.service                 enabled          enabled

~ A Pair Ctrl | ⌘ Dispatch Beta Ctrl Shift |
systemctl list-unit-files --type=service --state=running (tab)
```

Servicios activos al iniciar el sistema operativo

```
31 de may 23:59
systemctl show cron
Type=simple
ExitType=main
Restart=on-failure
RestartMode=normal
NotifyAccess=none
RestartUsec=100ms
RestartStep=0
RestartMaxDelayUsec=infinity
RestartUsecNext=100ms
TimeoutStartUsec=1min 30s
TimeoutStopUsec=1min 30s
TimeoutAbortUsec=1min 30s
TimeoutStartFailureMode=terminate
TimeoutStopFailureMode=terminate
RuntimeMaxUsec=infinity
RuntimeRandomizedExtraUsec=0
WatchdogUsec=0
WatchdogTimestampMonotonic=0
RootDirectoryStartOnly=no
RemainAfterExit=no
GuessMainPID=yes
MainPID=1093
ControlPID=0
FileDescriptorStoreMax=0
NFileDescriptorStore=0
FileDescriptorStorePreserve=restart
StatusErrno=0
Result=success
ReloadResult=success
CleanResult=success
UID=[not set]
GID=[not set]
NRestarts=0
OOMPolicy=stop
ReloadSignal=1
ExecMainStartTimestamp=Sat 2025-05-31 10:19:55 -03
ExecMainStartTimestampMonotonic=3940393
ExecMainExitTimestampMonotonic=0
ExecMainPID=1093
ExecMainCode=0
ExecMainStatus=0
```

Información detallada del proceso cron.service

4. Análisis.

1. ¿Qué similitudes y diferencias existen entre los servicios de Windows y los de Linux en cuanto a funcionamiento y parámetros?

Tanto en Windows como en Linux, los servicios (daemons en Linux) cumplen una función similar: ejecutan procesos en segundo plano que son esenciales para el funcionamiento del sistema o de determinadas aplicaciones.

Las diferencias principales radican en cómo se gestionan.

En Windows, los servicios son administrados a través del Service Control Manager, y pueden gestionarse mediante PowerShell. En Linux la gestión se realiza generalmente a través de systemd, utilizando comandos systemctl, la configuración de los daemons se realiza a través de archivos .service ubicados en directorios específicos del sistema /etc

2. ¿Cómo afecta la configuración de un servicio o daemon en los logs de eventos de cada sistema operativo?

La forma en que está configurado un servicio puede influir directamente en los logs del sistema. En Windows, cualquier cambio en el estado de un servicio queda registrado en el visor de eventos, dentro del log del sistema. Si un servicio está mal configurado o falla al

iniciarse, se generan eventos con distintos niveles de severidad, como "Error" o "Warning", junto con un ID de evento que ayuda a identificar el problema.

En Linux, los eventos de los daemons se registran a través del sistema de logs de el comando `journalctl`, y también se puede observar todo lo relacionado con el estado y los errores de un servicio. Una configuración incorrecta, como una ruta mal definida en el archivo `.service`, puede generar errores en el log, dificultar la ejecución del servicio o causar su reinicio constante.

En ambos sistemas operativos es fundamental una configuración adecuada y monitoreo para el correcto funcionamiento de los servicios

3. ¿Qué tipos de eventos generan los servicios en Windows frente a los daemons en Linux?

Los servicios en Windows generan eventos como el inicio y la detención del servicio, errores en la ejecución, fallos al cargar dependencias y eventos informativos relacionados con su funcionamiento. Estos eventos son categorizados como Information, error o warning en el visor de eventos, y se les asigna ID que ayuda en la administración y el diagnóstico.

En Linux, los daemons generan eventos similares, Estos eventos se registran en los logs del sistema `journalctl`, y se les asigna un nivel de severidad

En ambos casos, los eventos permiten monitorear el comportamiento del servicio y son claves para identificar problemas.

4. ¿Cómo influyen los parámetros de inicio automático en el rendimiento general del sistema en ambos casos?

Tanto en Windows como en Linux, los servicios que están configurados para iniciarse automáticamente al encender el sistema afectan directamente el tiempo de arranque y el consumo de recursos. Cuantos más servicios se inician al arranque, mayor será la carga inicial del sistema operativo y más recursos estarán ocupados desde el comienzo.

En Windows, algunos servicios innecesarios pueden quedar habilitados por defecto y ralentizar el sistema si no se gestionan adecuadamente, los puedes gestionar desde el task management.

En Linux, aunque el sistema suele ser más modular y optimizado, también es posible que ciertos daemons estén habilitados sin necesidad, especialmente en sistemas que usan entornos gráficos pesados o distribuciones preconfiguradas.

5. ¿Qué desafíos surgen al administrar servicios en cada sistema operativo?

En Windows, uno de los principales desafíos es la dependencia entre servicios. A veces detener un servicio puede afectar el funcionamiento de otros que dependen de él. Además, algunos errores reportados por los servicios se muestran mediante códigos que no siempre son claros.

En Linux, la administración de daemons puede ser más compleja, especialmente porque requiere conocer bien el funcionamiento de systemd, cada comando y atributo del mismo, así como editar correctamente archivos de configuración. Errores de sintaxis en un archivo de configuración pueden impedir que el servicio arranque o incluso que el sistema inicie correctamente si se trata de un servicio crítico.

Ambos sistemas presentan sus particularidades, pero en general, la administración de servicios en Linux tiende a ser más flexible y potente, aunque también más técnica. Tengo que reconocer que en Linux se me complicó más para visualizar todos los eventos, daemons, y comprender archivos de configuración. Mientras que en Windows es más accesible, didáctico y visual para usuarios sin conocimiento técnico.