

# Reto Día 6: Configuración Avanzada de Windows Server

Por Daniel Ariza

17/06/2025

## Fase 1: Preparación del entorno y consola administrativa

- Crear un **usuario administrador secundario** con una contraseña compleja.

```
PS C:\Users\daniel> net user admin2 Mech@tr3 /add
Se ha completado el comando correctamente.

PS C:\Users\daniel> _
```

Con este comando creamos un administrador secundario y su contraseña segura.

- Configurar una directiva de seguridad local para que las contraseñas caduquen cada 30 días.

```
C:\Administrador: C:\Windows\system32\cmd.exe
Se ha completado el comando correctamente.

PS C:\Users\daniel> net accounts
Tiempo antes del cierre forzado:          Nunca
Duración mín. de contraseña (días):       0
Duración máx. de contraseña (días):       42
Longitud mínima de contraseña:            0
Duración del historial de contraseñas:     Ninguna
Umbral de bloqueo:                        Nunca
Duración de bloqueo (minutos):             30
Ventana de obs. de bloqueo (minutos):     30
Rol del servidor:                         SERVIDOR
Se ha completado el comando correctamente.

PS C:\Users\daniel>
```

Con este comando verificamos cual es el tiempo actual de duración de contraseñas. Actualmente es de 42 días.

```
PS C:\Users\daniel> net accounts /maxpwage:30
Se ha completado el comando correctamente.

PS C:\Users\daniel> net accounts
Tiempo antes del cierre forzado:          Nunca
Duración mín. de contraseña (días):      0
Duración máx. de contraseña (días):      30
Longitud mínima de contraseña:           0
Duración del historial de contraseñas:     Ninguna
Umbral de bloqueo:                        Nunca
Duración de bloqueo (minutos):            30
Ventana de obs. de bloqueo (minutos):     30
Rol del servidor:                         SERVIDOR
Se ha completado el comando correctamente.

PS C:\Users\daniel>
```

Con el siguiente comando actualizamos la duración de las contraseñas a 30 días y comprobamos que el cambio se ha realizado correctamente.

- Cambiar la configuración del Control de Cuentas de Usuario (UAC) para mayor control de privilegios.

```
PS C:\Users\daniel> set-itemproperty -path "hklm:/software/microsoft/windows/currentversion/policies/system" -name conse
ntpromptbehavioradmin -value 2
PS C:\Users\daniel> set-itemproperty -path "hklm:/software/microsoft/windows/currentversion/policies/system" -name local
accounttokenfilterpolicy -value 0
PS C:\Users\daniel> set-itemproperty -path "hklm:/software/microsoft/windows/currentversion/policies/system" -name enabl
elua -value 1
PS C:\Users\daniel>
```

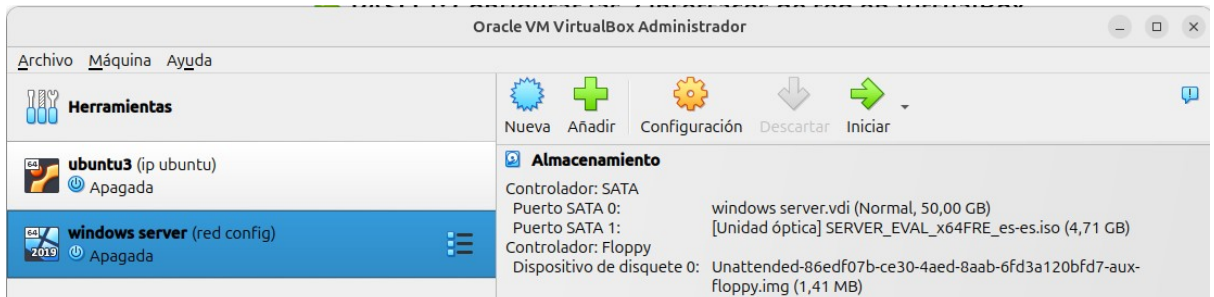
Con estos tres comandos buscamos que al iniciar sesión con un administrador tengamos más restricciones al ejecutar acciones elevadas.

Aplicar incluso un control más estricto al usuario administrador. Y hacer que los entornos de prueba y producción sean más seguros.

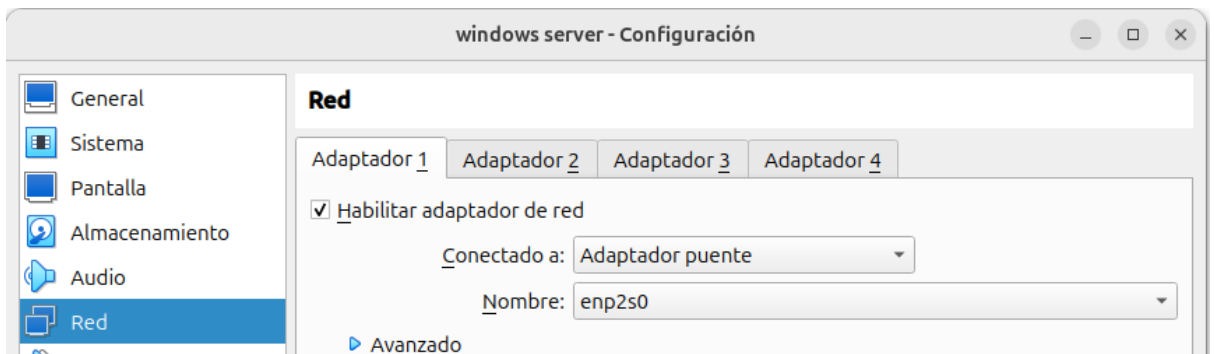
Para que estos cambios tengan efecto debemos reiniciar el sistema.

## Fase 2: Ajustes de red y servicios.

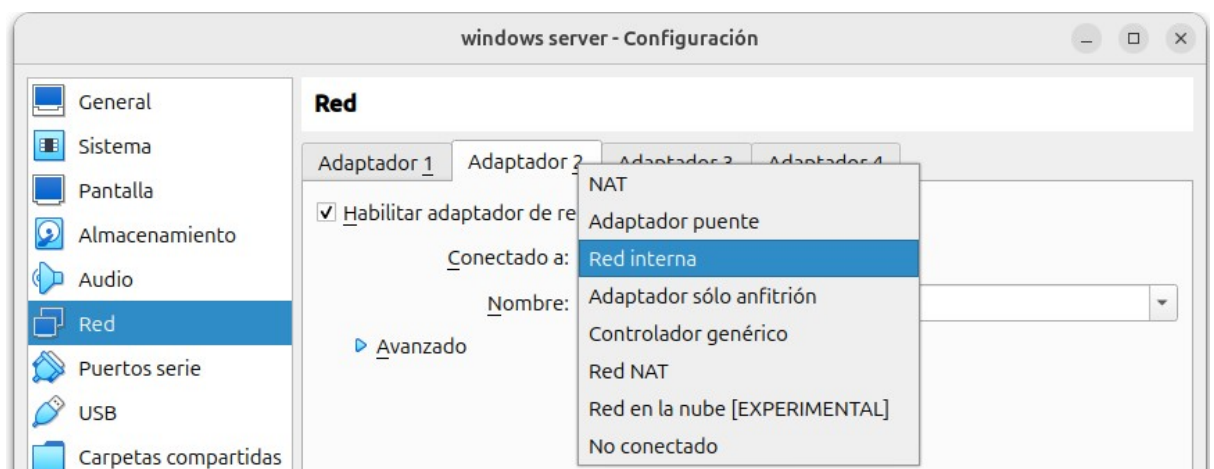
- Establecer **dos tarjetas de red** en la máquina virtual: una para conexión interna, otra para externa.



Apagamos nuestro Windows server y nos situamos en la configuración de red de nuestra máquina virtual en Virtual Box.



Vamos al apartado de red y nos aseguramos que esté habilitado el adaptador 1 y en la opción “adaptador puente”. Esta será nuestra conexión externa.



Marcamos la casilla de habilitar adaptador de red 2 y lo conectamos a “red interna” y le asignamos un nombre que en nuestro caso será “intnet”. Esta será nuestra conexión interna.

```
Administrador: C:\Windows\system32\cmd.exe
ADVERTENCIA: Para iniciar la herramienta de configuración del servidor de nuevo, ejecute "SConfig"
PS C:\Users\daniel> get-netadapter

Name                           InterfaceDescription           ifIndex Status      MacAddress           LinkSpeed
----                           -
Ethernet 2                     Intel(R) PRO/1000 MT Desktop Adapter #2 10 Up        08-00-27-89-C2-6E    1 Gbps
Ethernet                       Intel(R) PRO/1000 MT Desktop Adapter    3 Up        08-00-27-31-48-EE    1 Gbps

PS C:\Users\daniel>
```

Volvemos a encender Windows server y con el siguiente comando vemos que están los dos adaptadores de red establecidos.

- Configurar **rutas estáticas** en la tabla de red para simular un entorno más complejo.

```
PS C:\Users\daniel> new-netipaddress -interfacealias "ethernet 2" -ipaddress 192.168.100.1 -PrefixLength 24

IPAddress      : 192.168.100.1
InterfaceIndex : 10
InterfaceAlias : Ethernet 2
AddressFamily  : IPv4
Type           : Unicast
PrefixLength   : 24
PrefixOrigin   : Manual
SuffixOrigin   : Manual
AddressState   : Tentative
ValidLifetime  : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource   : False
PolicyStore    : ActiveStore

IPAddress      : 192.168.100.1
InterfaceIndex : 10
InterfaceAlias : Ethernet 2
AddressFamily  : IPv4
Type           : Unicast
PrefixLength   : 24
PrefixOrigin   : Manual
SuffixOrigin   : Manual
AddressState   : Invalid
ValidLifetime  : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource   : False
PolicyStore    : PersistentStore
```

Con este comando le asignaremos una ip estática a nuestro adaptador de red 2.

- Crear y activar un servidor DNS local, añadiendo una zona directa con al menos 2 registros.

```
CA. Administrador: C:\Windows\system32\cmd.exe

PS C:\Users\daniel> install-windowsfeature -name dns -includemanagementtools

Success Restart Needed Exit Code      Feature Result
-----
True      No                Success      {Servidor DNS}

PS C:\Users\daniel>
```

Instalamos el servidor DNS.

```
PS C:\Users\daniel> add-dnsserverprimaryzone -name "laboratorio.local" -zonefile "laboratorio.local.dns" -dynamicupdate none
PS C:\Users\daniel>
```

Este comando nos creará una zona primaria sin actualizaciones dinámicas. (ideal para pruebas). La zona la hemos llamado laboratorio.local

```
PS C:\Users\daniel> add-dnsserverprimaryzone -name "laboratorio.local" -zonefile "laboratorio.local.dns" -dynamicupdate none
PS C:\Users\daniel> add-dnsserverresourcerecord -name "winserver" -zonename "laboratorio.local" -ipv4address 192.168.100.1
PS C:\Users\daniel> add-dnsserverresourcerecord -name "ubuntuserver" -zonename "laboratorio.local" -ipv4address 192.168.100.2
PS C:\Users\daniel>
```

Con estos dos comandos creamos dos registros "A" en la zona primaria anterior.

```
PS C:\Users\daniel> nslookup winserver.laboratorio.local
DNS request timed out.
    timeout was 2 seconds.
Servidor: UnKnown
Address: 192.168.100.1

Nombre: winserver.laboratorio.local
Address: 192.168.100.1
```

Y con este comando comprobamos que el servidor DNS local funciona correctamente.

## Fase 3: Personalización del entorno de trabajo del servidor.

- Habilitar el Escritorio Remoto y limitar el número de sesiones a 2.

```
CA. Administrador: C:\Windows\system32\cmd.exe

ADVERTENCIA: Para iniciar la herramienta de configuración del servidor de nuevo, ejecute "SConfig"
PS C:\Users\daniel> reg add "hklm\system\currentcontrolset\control\terminal server" /v fdenytsconnections /t reg_dword /d 0 /f
La operación se completó correctamente.
PS C:\Users\daniel>
```

Con este comando habilitamos el escritorio remoto (RDP).

```
PS C:\Users\daniel> netsh advfirewall firewall set rule group="escritorio remoto" new enable=yes
Se actualizaron 3 reglas.
Aceptar
PS C:\Users\daniel>
```

Y con este otro comando abrimos el puerto 3389 en el firewall que hará que permita establecer las conexiones remotas a nuestro servidor.

```
El grupo local especificado no existe.
PS C:\Users\daniel> net localgroup "usuarios de escritorio remoto" daniel /add
Se ha completado el comando correctamente.
PS C:\Users\daniel>
```

introduciendo este otro comando añadimos al usuario "daniel" al grupo de "usuarios de escritorio remoto".

```
Se ha completado el comando correctamente.
PS C:\Users\daniel> reg add "hkLM\system\currentcontrolset\control\terminal server" /v maxinstancecount /t reg_dword /d 2 /f
La operación se completó correctamente.
PS C:\Users\daniel>
```

Ahora introducimos este comando para limitar las sesiones a 2. Hay que reiniciar el servidor para que se apliquen los cambios.

```
PS C:\Users\daniel> query session

```

NOMBRE DE SESIÓN	NOMBRE DE USUARIO	ID	ESTADO	TIPO	DISPOSITIVO
services		0	Desc		
>console	daniel	1	Activo		
rdp-tcp		65536	Escuchar		

```
PS C:\Users\daniel>
```

Con este otro comando después de reiniciar el servidor vemos que todo funciona correctamente.

```
especificados.
PS C:\Users\daniel> reg query "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v MaxInstanceCount

```

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server			
MaxInstanceCount	REG_DWORD	0x2	

```
PS C:\Users\daniel>
```

Y vemos también que el número de sesiones se ha limitado a 2 correctamente.

- Personalizar el inicio del sistema añadiendo un script que cree automáticamente una carpeta de logs en C:\Logs.

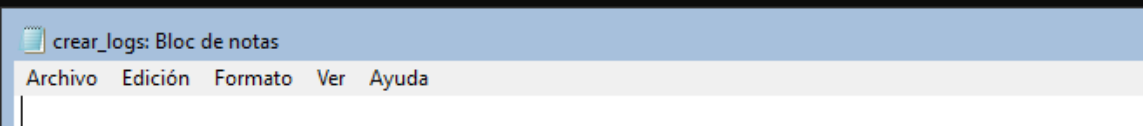
```
PS C:\Users\daniel> New-item -path "C:\scripts" -ItemType Directory

Directorio: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          17/06/2025     8:42             scripts
```

Con este comando crearemos la carpeta donde alojaremos el script

```
PS C:\Users\daniel> notepad C:\scripts\crear_logs.ps1
PS C:\Users\daniel> _
```

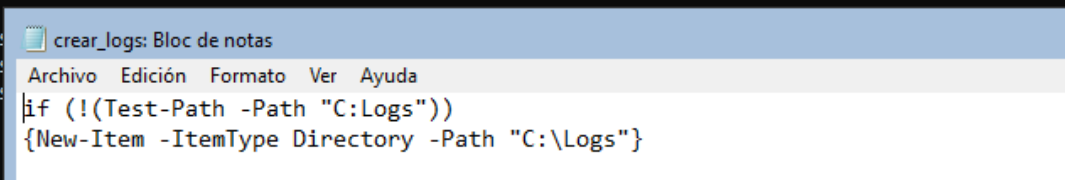


Con este otro comando creamos el archivo del script.

```
Administrator: C:\Windows\system32\cmd.exe

Mode                LastWriteTime         Length Name
----                -
d-----          17/06/2025     8:42             scripts

PS C:\Users\daniel> notepad C:\scripts\crear_logs.ps1
PS C:\Users\daniel> _
PS C:\Users\daniel> if (!(Test-Path -Path "C:\Logs"))
{New-Item -ItemType Directory -Path "C:\Logs"}
```



Dentro del archivo escribiremos esto. Quiere decir lo siguiente: si no existe la carpeta C:\Logs, creala.

Usaremos el programador de tareas de Windows Server desde Power Shell y ejecutaremos los siguientes comandos:

```
PS C:\Users\daniel> $accion = New-ScheduledTaskAction -Execute "powershell.exe" -Argument "ExecutionPolicy Bypass -File C:\scripts\crear_logs.ps1"
PS C:\Users\daniel> $disparador = New-ScheduledTaskTrigger -AtStartup
PS C:\Users\daniel> Register-ScheduledTask -TaskName "CrearCarpetaLogs" -Action $accion -Trigger $disparador -RunLevel Highest -User "SYSTEM"

TaskPath                TaskName                State
-----
\                        CrearCarpetaLogs        Ready

PS C:\Users\daniel>
```

Esto crea una tarea que se ejecuta al arrancar Windows utilizando la Power Shell que ejecuta el script crear\_logs.ps1 con permisos del sistema.

```
PS C:\Users\daniel> Get-Item C:\Logs

Directorio: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          17/06/2025   10:29             Logs

PS C:\Users\daniel>
```

Con este comando comprobamos que la carpeta se crea correctamente.

- Configurar el firewall para que sólo permita el tráfico RDP y DNS.

```
PS C:\Users\daniel> netsh advfirewall set allprofiles firewallpolicy blockinboundalways,allowoutbound
Aceptar
PS C:\Users\daniel> _
```

Con este comando bloqueamos todo el tráfico entrante y permitimos todo el tráfico saliente.

```
PS C:\Users\daniel> netsh advfirewall firewall add rule name="Permitir RDP" dir=in action=allow protocol=TCP localport=3389
Aceptar
PS C:\Users\daniel> _
```

Esto evita que nos bloqueemos a nosotros mismos cuando usamos RDP para trabajar con el servidor.

```
PS C:\Users\daniel> netsh advfirewall firewall add rule name="Permitir DNS TCP" dir=in action=allow protocol=TCP localport=53
Aceptar
PS C:\Users\daniel> netsh advfirewall firewall add rule name="Permitir DNS UDP" dir=in action=allow protocol=UDP localport=53
Aceptar
```

Con esto el servidor podrá resolver consultas DNS o resolver dominios desde clientes.



```

C:\Windows\system32\cmd.exe
Aceptar
PS C:\Users\daniel> netsh advfirewall firewall show rule name=all | findstr "RDP DNS"
Nombre de regla: Permitir DNS UDP
Nombre de regla: Permitir DNS TCP
Nombre de regla: Permitir RDP
Agrupamiento: Servicio DNS
Nombre de regla: DNS (TCP, entrantes)
Agrupamiento: Servicio DNS
Agrupamiento: Servicio DNS
Agrupamiento: Servicio DNS
Nombre de regla: DNS (UDP, entrantes)
Agrupamiento: Servicio DNS
Agrupamiento: Servicio DNS
Nombre de regla: mDNS (UDP de entrada)
Agrupamiento: mDNS
Nombre de regla: mDNS (UDP de entrada)
Agrupamiento: mDNS
Nombre de regla: mDNS (UDP de salida)
Agrupamiento: mDNS
Nombre de regla: mDNS (UDP de entrada)
Agrupamiento: mDNS
Nombre de regla: Redes principales: DNS (UDP de salida)
Nombre de regla: mDNS (UDP de salida)
Agrupamiento: mDNS
Nombre de regla: mDNS (UDP de salida)
Agrupamiento: mDNS
PS C:\Users\daniel>

```

Y para terminar verificamos que todo se ha implementado con éxito. La captura demuestra que el servidor **permite correctamente tráfico RDP y DNS**.

## Fase 4: Automatización básica.

- Crear un script en PowerShell que realice las siguientes tareas:
  - Cree una carpeta con la fecha actual.
  - Copie archivos del escritorio a esa carpeta.
  - Genere un log en .txt con el resultado de la copia.

```
PS C:\Users\daniel> notepad C:\scripts\backup_escritorio.ps1
PS C:\Users\daniel>

Archivo Edición Formato Ver Ayuda
# Obtener fecha actual en formato yyyy-MM-dd
$fecha = Get-Date -Format "yyyy-MM-dd"

# Ruta destino: C:\Backups\yyyy-MM-dd
$destino = "C:\Backups\$fecha"

# Crear la carpeta si no existe
if (!(Test-Path -Path $destino))
{New-Item -ItemType Directory -Path $destino}

# Ruta del escritorio del usuario actual
$escritorio = [Environment]::GetFolderPath("Desktop")

# Ruta del log
$log = "$destino\resultado_copia.txt"

# Copiar archivos del escritorio al destino y guardar salida en log
Copy-Item "$escritorio\*" -Destination $destino -Recurse -Force -ErrorAction SilentlyContinue -Verbose *>1 | Out-File $log
```

Creamos este archivo y escribimos el siguiente script. Lo ejecutamos.

```
PS C:\Users\daniel> Get-ChildItem C:\Backups

Directorio: C:\Backups

Mode                LastWriteTime         Length Name
----                -
d-----           17/06/2025   13:27             2025-06-17

PS C:\Users\daniel> _
```

Podemos ver como se ha creado la carpeta con la fecha.

```
PS C:\Users\daniel> Get-ChildItem "C:\Backups\2025-06-17"

Directorio: C:\Backups\2025-06-17

Mode                LastWriteTime         Length Name
----                -
d-----           17/06/2025   13:23             $WinREAgent
d-----           17/06/2025   13:23             Archivos de programa
-a-----           17/06/2025   13:27             318 resultado_copia.txt
```

Vemos como los archivos también se han copiado. El archivo **resultado\_copia.txt** con 318 bytes → esto confirma que el **log del proceso de copia** también se ha generado.

A continuación adjunto el código del script:

```
# Obtener fecha actual en formato yyyy-MM-dd
```

```
$fecha = Get-Date -Format "yyyy-MM-dd"
```

```
# Ruta destino: C:\Backups\yyyy-MM-dd
```

```
$destino = "C:\Backups\$fecha"
```

```
# Crear la carpeta si no existe
```

```
if (!(Test-Path -Path $destino)) {
```

```
    New-Item -ItemType Directory -Path $destino
```

```
}
```

```
# Ruta del escritorio del usuario actual
```

```
$escritorio = [Environment]::GetFolderPath("Desktop")
```

```
# Ruta del log
```

```
$log = "$destino\resultado_copia.txt"
```

```
# Copiar archivos del escritorio al destino y guardar salida en log
```

```
Copy-Item "$escritorio\*" -Destination $destino -Recurse -Force -  
ErrorAction SilentlyContinue -Verbose *>&1 | Out-File $log
```