



Configuración de Windows Server 2025: Administración, Red, Seguridad y Automatización

16/06/2025

Angel Moreno Garcia

CodeArts

Madrid

Introducción

Configuración y optimización de un entorno de servidor Windows con buenas prácticas de seguridad, administración avanzada, configuración de red y automatización de tareas.

Objetivos

1. Configurar un servidor Windows en un entorno virtualizado.
2. Aplicar directivas de seguridad y control de acceso.
3. Optimizar la conectividad de red.
4. Automatizar tareas de administración con PowerShell.

Descripción del entorno de trabajo

- Hypervisor: Para la virtualización de las máquinas se ha empleado **VirtualBox**.
- ISO del sistema: La ISO del sistema es **Windows Server** 2025 (versión de escritorio).
- Configuración hardware de la VM:
 - **CPU:** 3 núcleos
 - **RAM:** 4 GB
 - **Disco:** 50 GB
 - Adaptadores de red: Tipo **Bridge**

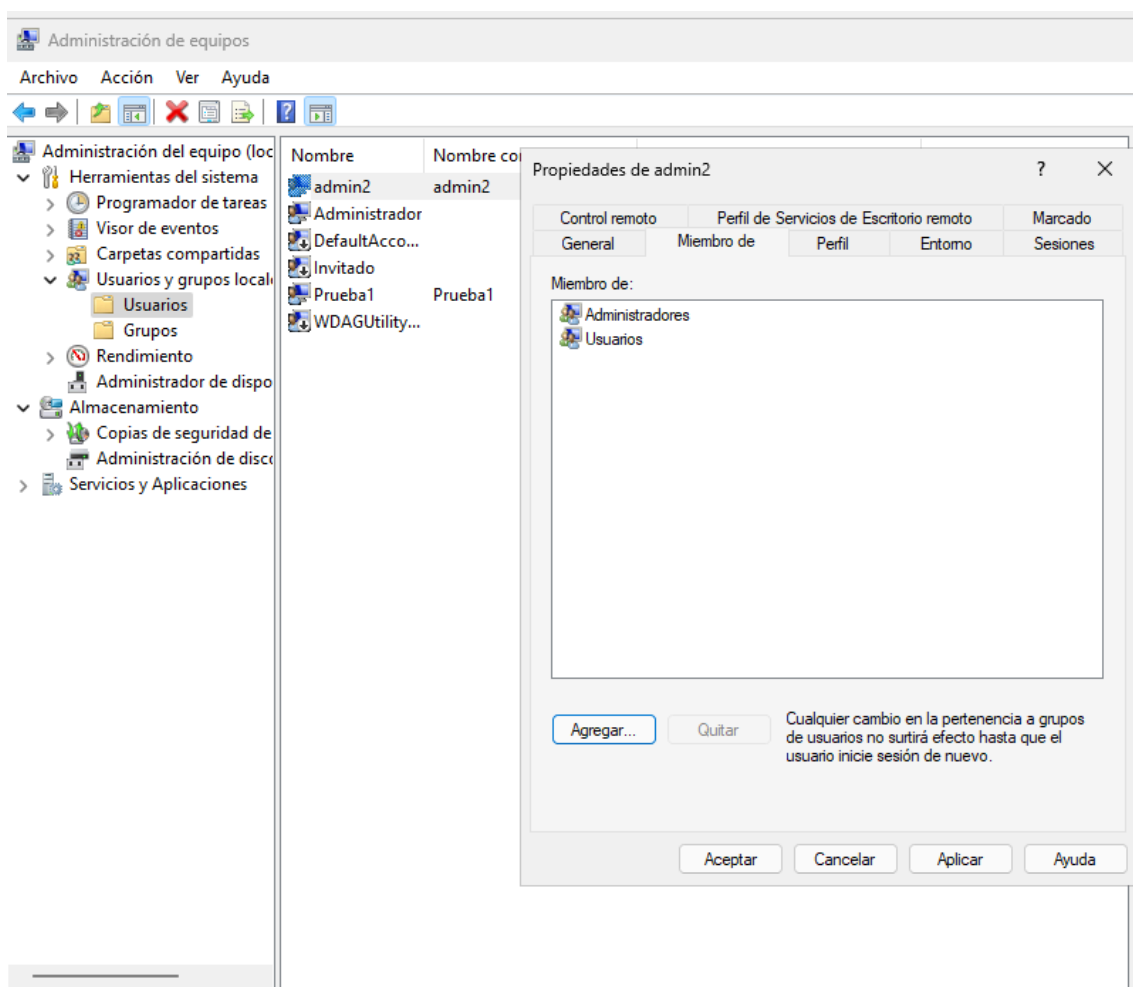
4. Fase 1: Preparación del entorno y consola administrativa

I. Creación de usuario administrador secundario

Usuario: **admin2**

Contraseña: **contraseña compleja**

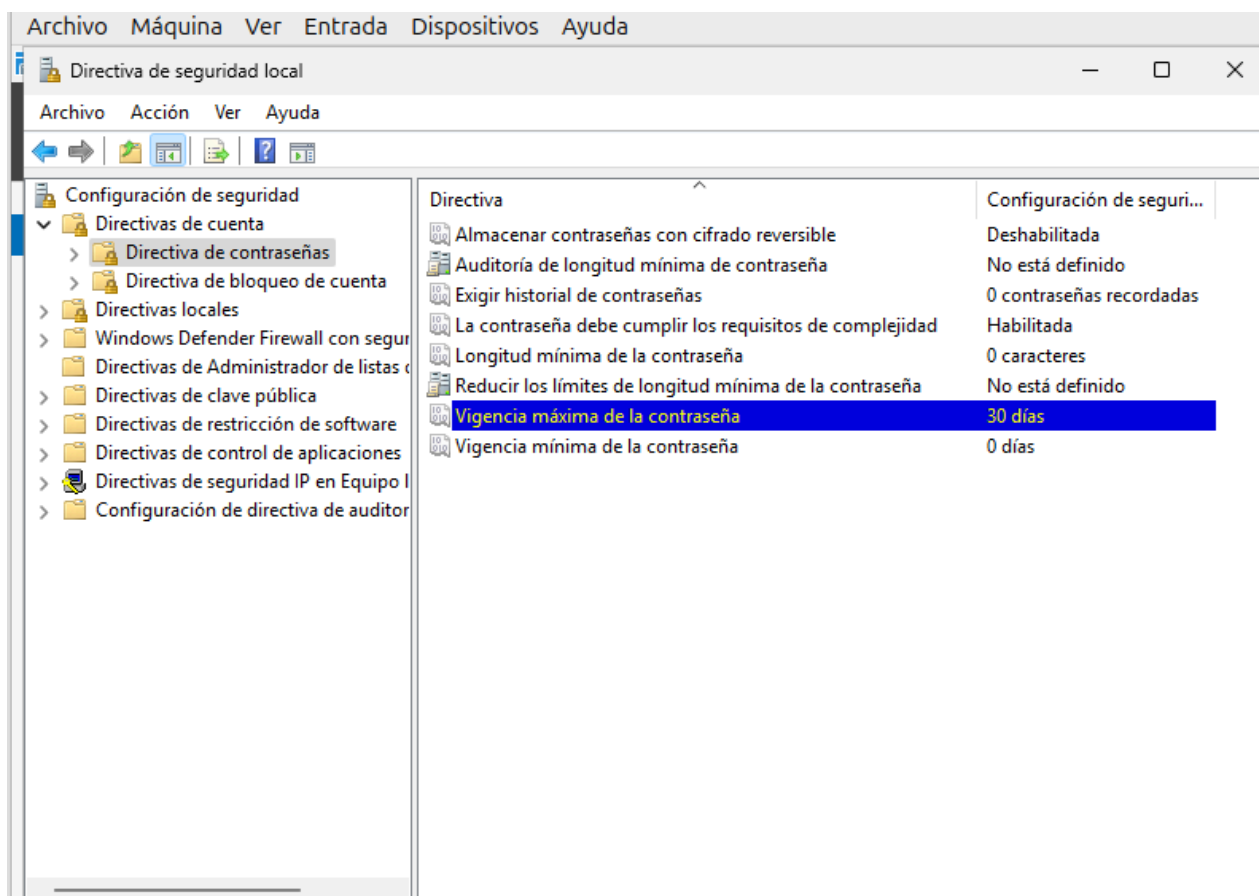
Método utilizado: Usuarios y grupos locales (lusrmgr.msc)



II. Configuración de directiva de caducidad de contraseñas

Valor configurado: **30 días**

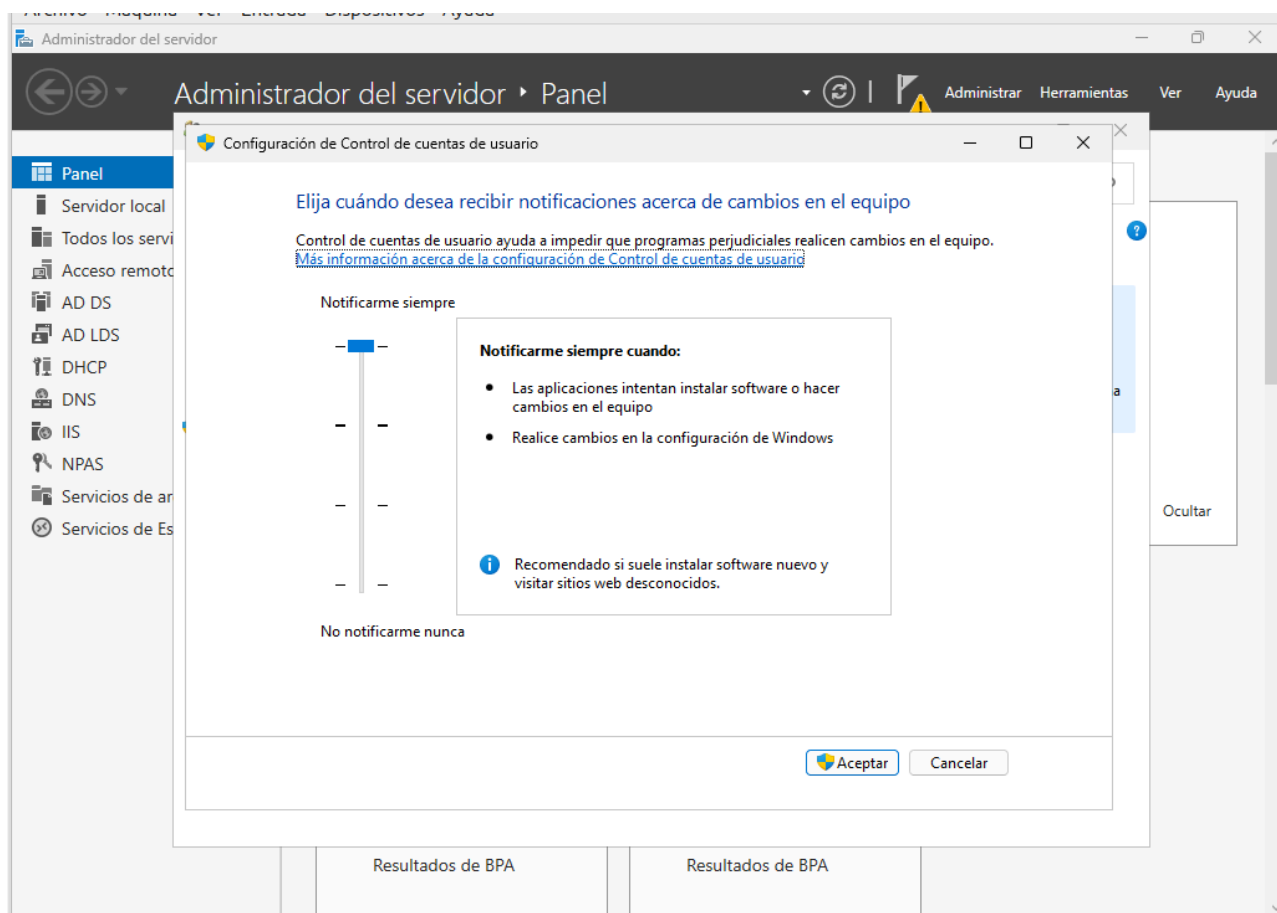
Ruta: **secpol.msc** → **Directivas de cuenta** → **Directiva de contraseñas**



III. Configuración del Control de Cuentas de Usuario (UAC)

Nivel configurado: **máximo control**

Ruta: **Panel de control** → **Cuentas de usuario** → **Cambiar configuración de UAC**



5. Fase 2: Ajustes de red y servicios

I. Configuración del Control de Cuentas de Usuario (UAC)

Adaptador 1: Interna (red interna).

Adaptador 2: Se utilizó un servidor Ubuntu para la complejidad de la prueba.

IP asignadas:

Propiedades de Protocolo de Internet versión 4 (TCP/IPv4)

General

Puede hacer que la configuración IP se asigne automáticamente si la red es compatible con esta funcionalidad. De lo contrario, deberá consultar con el administrador de red cuál es la configuración IP apropiada.

☐ Obtener una dirección IP automáticamente

☒ Usar la siguiente dirección IP:

Dirección IP: 192 . 168 . 1 . 138

Máscara de subred: 255 . 255 . 255 . 0

Puerta de enlace predeterminada: 192 . 168 . 1 . 1

☐ Obtener la dirección del servidor DNS automáticamente

☒ Usar las siguientes direcciones de servidor DNS:

Servidor DNS preferido: 8 . 8 . 8 . 8

Servidor DNS alternativo: 8 . 8 . 4 . 4

☐ Validar configuración al salir

Opciones avanzadas...

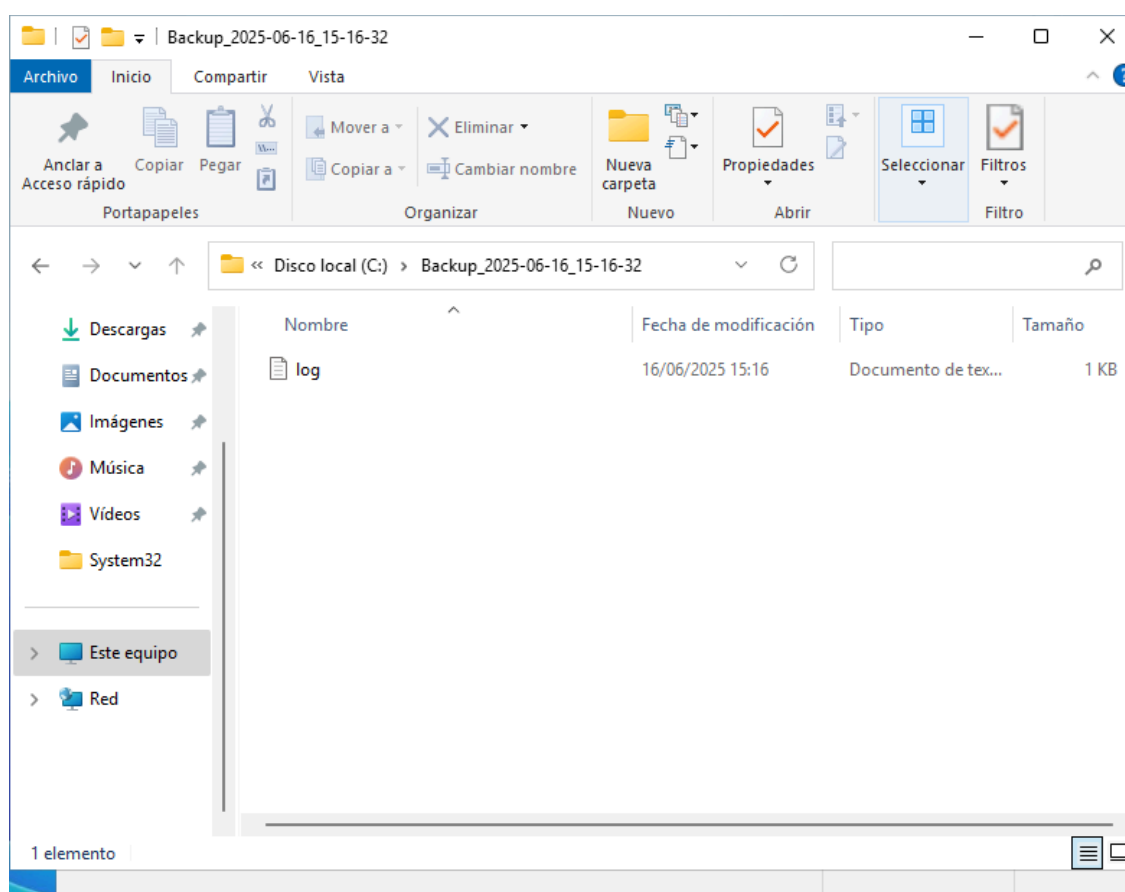
Aceptar Cancelar

6. Fase 3: Personalización del entorno de trabajo

I. Script de inicio para crear carpeta C:\Logs

Tipo de script: **ps1**.

Ubicación: **C:\Backup_YYYY_MM_DD**



Script en Powershell:

```

Administrador: Windows Pow  X  +  v
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\Administrador> # Obtener la fecha actual para nombrar la carpeta
PS C:\Users\Administrador> $fechaActual = Get-Date -Format "yyyy-MM-dd_HH-mm-ss"
PS C:\Users\Administrador>
PS C:\Users\Administrador> # Definir la ruta del escritorio y la carpeta de destino
PS C:\Users\Administrador> $escritorio = [System.Environment]::GetFolderPath('Desktop')
PS C:\Users\Administrador> $directorioDestino = "C:\Backup_{$fechaActual}"
PS C:\Users\Administrador>
PS C:\Users\Administrador> # Crear la carpeta de destino si no existe
PS C:\Users\Administrador> if (-not (Test-Path $directorioDestino)) {
>>     New-Item -Path $directorioDestino -ItemType Directory
>> }

Directorio: C:\

Mode                LastWriteTime         Length Name
----                -
d-----         16/06/2025   15:16             Backup_2025-06-16_15-16-32

PS C:\Users\Administrador>
PS C:\Users\Administrador> # Definir el archivo de log
PS C:\Users\Administrador> $logFile = "C:\Backup_{$fechaActual}\log.txt"

```

```

Administrador: Windows Pow  X  +  v
PS C:\Users\Administrador> function Copiar-Archivos {
>>     param (
>>         [string]$origen,
>>         [string]$destino
>>     )
>>
>>     # Obtener los archivos en el escritorio
>>     $archivos = Get-ChildItem -Path $origen
>>
>>     # Comenzar a escribir en el archivo de log
>>     Add-Content -Path $logFile -Value "Inicio de la copia: $(Get-Date)"
>>
>>     foreach ($archivo in $archivos) {
>>         try {
>>             # Copiar el archivo
>>             $destinoArchivo = Join-Path -Path $destino -ChildPath $archivo.Name
>>             Copy-Item -Path $archivo.FullName -Destination $destinoArchivo -Force
>>
>>             # Registrar éxito en el log
>>             Add-Content -Path $logFile -Value "Copia exitosa: $($archivo.FullName) a $destinoArchivo"
>>         } catch {
>>             # Registrar error en el log
>>             Add-Content -Path $logFile -Value "Error al copiar el archivo: $($archivo.FullName) - $_"
>>         }
>>     }
>>
>>     # Finalizar con un mensaje en el log
>>     Add-Content -Path $logFile -Value "Fin de la copia: $(Get-Date)"
>> }
PS C:\Users\Administrador>

```


Ruta: **gpedit.msc** → **Configuración de Windows** → **Scripts** → **Agregar**

