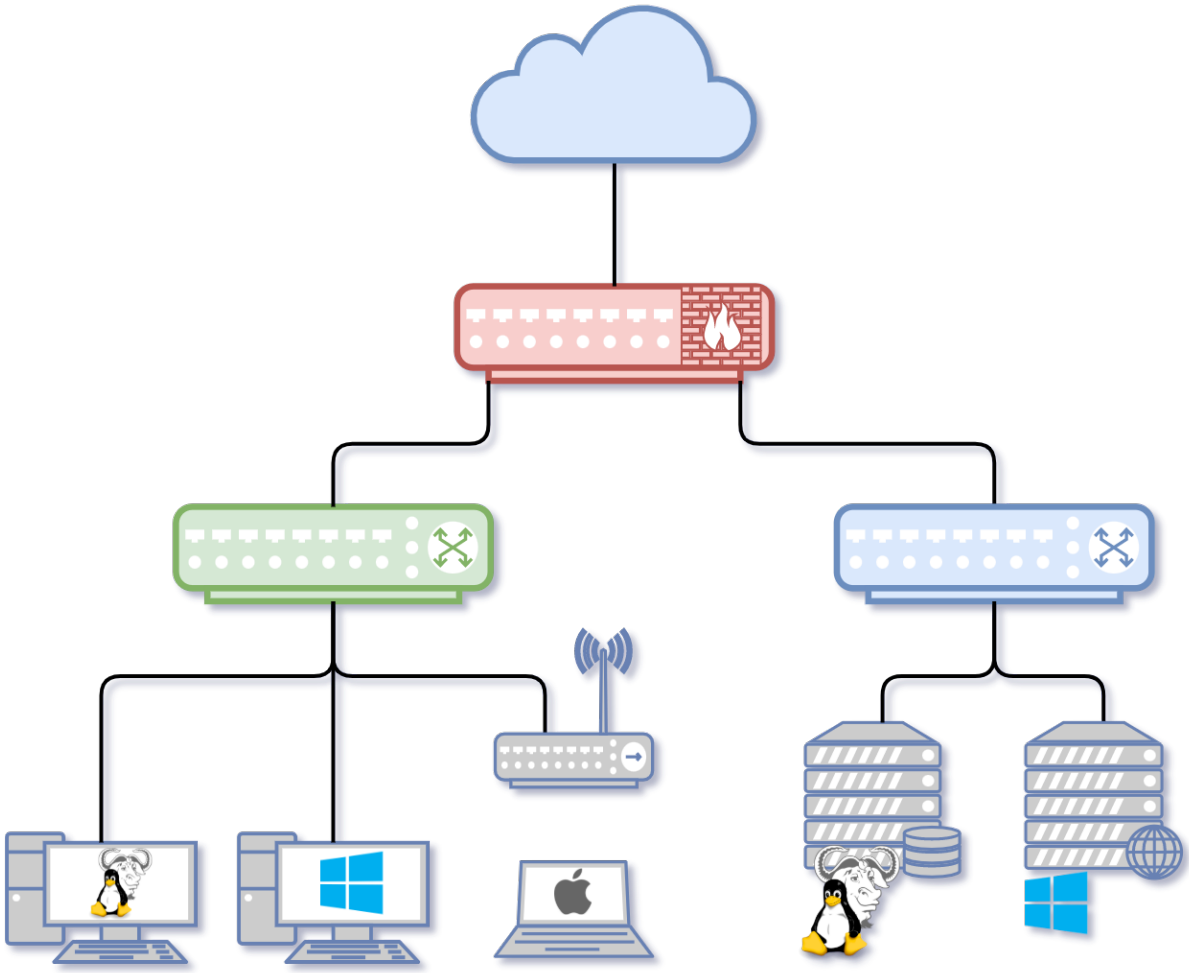
Técnico superior en Administración de Sistemas informáticos en red



Administración de Sistemas Operativos

Rubén Gómez Olivencia

2022-2023

- Github: https://github.com/yuki

Puedes encontrar la última versión de este libro en formato **HTML** en el siguiente link, así como otros libros que he creado. Para descargar el código fuente en formato **Markdown** visita el repositorio en GitHub.

Información

Por favor, ponte en contacto conmigo si encuentras algún fallo, falta de ortografía o quieres mejorar de alguna manera este libro. Gracias.

I Windows Server 2022

Windows Server 2022

1. Un poco de historia

Windows Server es la línea de productos de Microsoft **enfocada a servidores** que se inició con la primera versión: Windows Server 2003.

Anteriormente, Microsoft contaba con una línea también dedicada a estaciones de trabajo y servidores en red cuyo nombre era Windows NT, por lo que se puede considerar que **Windows Server es la continuación de NT** con el cambio de nombre.

2. Proceso de instalación de Windows Server 2022

Para realizar la instalación de Windows Server 2022 necesitaremos el medio desde el que realizaremos la instalación. Microsoft permite la descarga desde su página oficial una evaluación de 180 días que podremos descargar en varios formatos:

- **Azure**: Es el sistema "en la nube" de Microsoft. Se puede probar Windows Server a través de una cuenta gratuita y posteriormente gestionar los sistemas virtualizados que se creen en la nube.
- **ISO**: Una imagen ISO que podremos utilizar grabándolo en un DVD, a través de un USB o usarlo en un sistema de virtualización propio.
- **VHD**: Formato de archivo que representa una unidad de disco duro virtual.

Para realizar una instalación completa en nuestro sistema de virtualización elegiremos el archivo ISO. Para poder descargarlo tendremos que rellenar un formulario, elegir el idioma y posteriormente comenzará la descarga.

2.1. Requisitos previos

Antes de realizar la instalación debemos conocer cuáles son los requisitos mínimos de hardware que necesita Windows Server para así asegurar que la máquina virtual (o el hardware donde lo vamos a instalar) es 100 % compatible. En este caso, y tal como nos indica la web de Microsoft, los requisitos son:

- **Procesador** de 64 bits a 1,4 GHz, compatible con el conjunto de instrucciones x64
  - Admite DEP y NX
  - Admite CMPXCHG16b, LAHF/SAHF y PrefetchW
  - Admite la traducción de direcciones de segundo nivel (EPT o NPT)
- **RAM**: 512 MB (**2 GB** para la opción de instalación Servidor con Experiencia de escritorio). Se admite RAM tipo ECC (código de corrección de errores)
- **Espacio en disco**: mínimo 32GB. Windows Server no admite discos ATA, PATA, IDE y EIDE para unidades de arranque, página o datos.
- **Adaptador de red** de 1 gigabit/s

Dado que estos son los requisitos mínimos, nuestra máquina virtual deberá cumplirlos, pero en un sistema que vaya a estar en producción deberemos realizar un análisis de requisitos para asegurar que el hardware (ya sea virtual o físico) cumple las necesidades de los servicios que alojará:

- ¿El servidor va a contar con un servidor web?
  - ¿Cuántas visitas se esperan?
  - ¿Qué tipo de web va a servir? ¿Programada en Java, PHP, …?
- ¿El servidor va a contar con un sistema gestor de base de datos?
  - ¿Cuántas bases de datos va a tener?
  - ¿Cuántas conexiones simultáneas esperamos?
  - ¿Cuántas peticiones estimamos que va a recibir?
- ¿El servidor va a servir para compartir ficheros?
  - ¿Cuántos usuarios van a acceder?
  - ¿Peticiones estimadas?
- …

Por tanto, antes de realizar la instalación, deberemos confirmar que conocemos para qué se va a utilizar el servidor, cuántos servicios se van a utilizar y la carga esperada.

2.2. Instalación de Windows Server 2022

El asistente de instalación de Windows Server nos va a llevar por una serie de pasos que detallaremos a continuación.

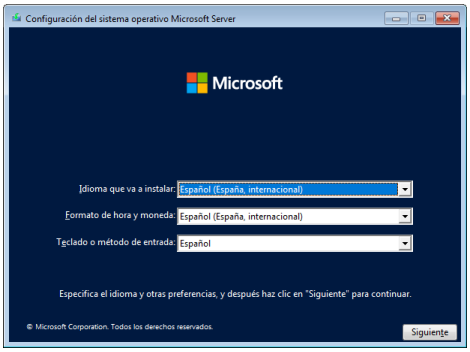
¡Atención!

Nos aseguramos que la máquina virtual arranque desde la ISO

2.2.1. Elección del idioma

El primer paso del asistente será elegir el idioma principal que usaremos en el sistema operativo. Antes de realizar la descarga de la ISO hemos elegido el idioma español, por lo que el asistente nos aparecerá por defecto en ese idioma. Podremos elegir:

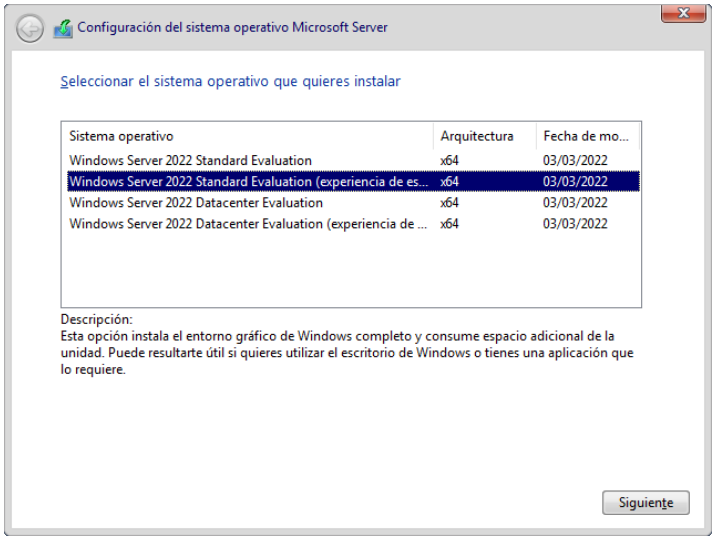
- Idioma que va a instalar
- Formato de hora y moneda
- Teclado o método de entrada



En nuestro caso dejaremos las opciones tal como nos aparece por defecto, pero de utilizar el servidor en un sistema internacional, convendría hacer uso del sistema en inglés.

2.2.2. Elección del sistema operativo

Windows Server 2022 tiene distintos modos a la hora de ser instalado, tal como podemos ver en la siguiente captura de pantalla del proceso de instalación:



Existen dos opciones principales a la hora de elegir, ya que cada una de ellas permitirá instalarlo con o sin experiencia de escritorio. Las opciones principales son:

- **Standard Evaluation**: Útil para empresas medianas o pequeñas, que no requieran de grandes sistemas de virtualización.
- **Datacenter Evaluation**: No habrá límite a la hora de crear máquinas virtuales con un host Hyper-V por cada licencia.

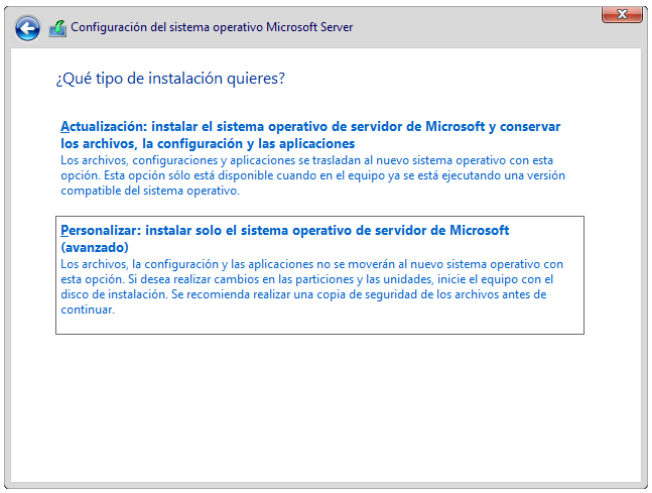
Existen más diferencias, y es por eso que Microsoft tiene una página dedicada con la comparación de ambas versiones. Por lo tanto, antes de realizar la instalación para nuestro sistema, debemos realizar una estimación de los requisitos para así poder elegir de manera adecuada.

Nosotros vamos a elegir la versión “Standard Evaluation” con experiencia de escritorio ya que podremos realizar la configuración desde el propio servidor. Dependiendo del caso, habrá que analizar cuál es la mejor opción antes de realizar la instalación.

Al darle a “Siguiente” nos aparecerán los términos de la licencia.

2.2.3. Tipo de instalación

Tras aceptar la licencia nos aparecerá el tipo de instalación que deseamos realizar:



- **Actualización**: Para realizar actualizaciones sobre sistemas compatibles ya instalados.
- **Personalizada**: Tal como indica la imagen anterior, los archivos, las configuraciones y las aplicaciones no se migran. En caso de realizar una instalación nueva (como en nuestro caso), **utilizaremos esta opción**.

Debemos tener en cuenta cuál es la situación en la que nos encontramos y decidir cuál es la mejor opción de las que se nos ofrece. Por eso es importante realizar la lectura de cada opción que nos aparece en todos los pasos de la instalación. Y más cuando se trata de la instalación de un sistema operativo de servidor

¡Atención!

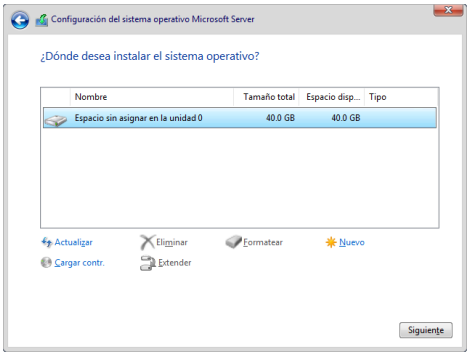
Es importante leer qué nos indica cada paso durante la instalación

2.2.4. Particionado de discos duros

Dado que nuestra máquina virtual es nueva, y no tiene ningún sistema operativo previo, vamos a tener que realizar la instalación en el disco duro que se ha añadido a la máquina virtual.

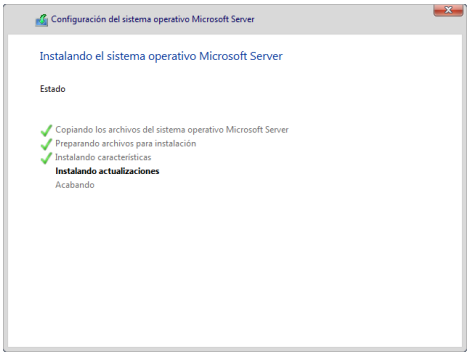
Dependiendo del tamaño que hayamos elegido, o incluso los discos duros que tengamos, nos aparecerán en la nueva pantalla del programa de instalación.

En nuestro caso la máquina virtual cuenta con un único disco duro de 40GB de almacenamiento, en donde realizaremos la instalación de la manera predeterminada que Windows Server elija particionarlo. Por lo tanto, seleccionaremos el disco duro y le daremos al botón de “Siguiente”.



2.2.5. Terminar el proceso de instalación

Tras haber seleccionado y haber completado todos los pasos que se han descrito hasta ahora, el proceso de instalación comenzará.

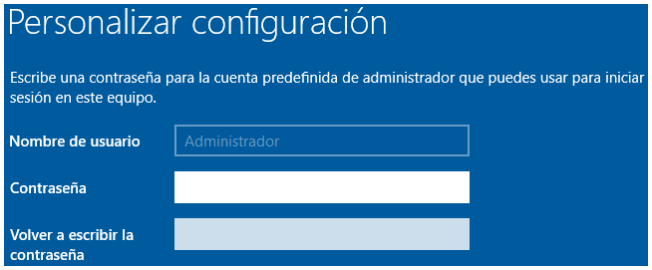


Este paso de la instalación requerirá de cierto tiempo que dependerá del hardware que tengamos disponible, ya que se realizará la copia de los ficheros de sistema, realizará las instalaciones pertinentes y por último instalará las actualizaciones necesarias.

2.2.6. Personalizar la configuración

Al terminar el proceso anterior, el último paso nos permitirá realizar la configuración de la contraseña del usuario **Administrador**.

Tal como se puede ver a continuación, el nombre de usuario no podrá ser modificado, y nos pedirá la contraseña y la confirmación de la contraseña.

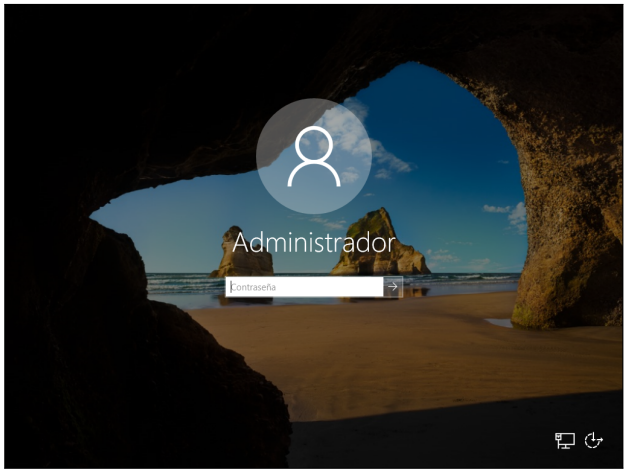


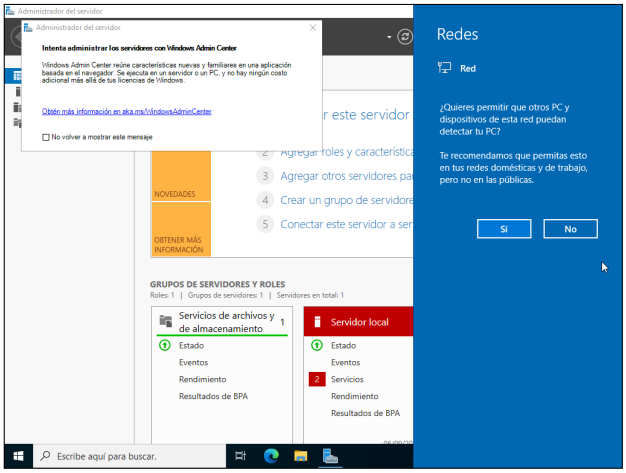
Cabe recordar que esta contraseña es muy importante, ya que con ella realizaremos la configuración de todo el sistema, y por tanto debe ser una contraseña segura y debemos guardarla a buen recaudo. Al darle a “Finalizar” el servidor se reiniciará.

2.3. Post-instalación

Una vez realizada la instalación conviene retirar la imagen ISO del sistema de virtualización, así como incluso retirar el lector DVD virtual, ya que en principio no será necesario volverlo a usar.

Tras el primer arranque nos aparecerá la siguiente pantalla para realizar el login con el usuario Administrador, en donde introduciremos la contraseña que hemos puesto en el paso anterior. Una vez introducida, la siguiente pantalla será la pantalla de inicio donde podremos ver el panel de administrador del servidor.





3. Windows Server 2022 como servidor de red

Windows Server 2022 tras la instalación no es más que un sistema operativo “normal”, que potencialmente se puede convertir en un Sistema Operativo de red, con funcionalidad para administrar usuarios, creación de grupos, permisos, … Para realizar estas funciones debemos de configurar el Sistema Operativo.

Como se ha observado, durante la instalación del Sistema Operativo apenas se realizan preguntas, por lo que es el propio instalador el que ha tomado ciertas decisiones por nosotros, como son:

- Obtención de IP (a través de DHCP o Dynamic Host Configuration Protocol)
- Nombre del equipo

Estos datos los modificaremos más adelante.

3.1. Entendiendo conceptos en Windows Server

Durante la instalación y promoción del servidor como controlador de dominio y configuración de Active Directory, existen ciertos conceptos que son intrínsecos a cómo va a funcionar Windows Server como servidor y que deben de ser entendidos.

3.1.1. Active Directory

Active Directory (AD), o **Directorio Activo**, son los términos que utiliza Microsoft para referirse a su implementación de servicio de directorio en una red distribuida de computadores. Utiliza distintos protocolos, principalmente LDAP, DNS, HDCP y Kerberos.

De forma sencilla se puede decir que es un servicio establecido en uno o varios servidores en donde se crean objetos tales como usuarios, equipos o grupos, con el objetivo de administrar los inicios de sesión en los equipos conectados a la red, así como también la administración de políticas en toda la red.

Su estructura jerárquica permite mantener una serie de objetos relacionados con componentes de una red, como usuarios, grupos de usuarios, permisos y asignación de recursos y políticas de acceso. (Fuente: wikipedia).

3.1.2. Dominio

Un dominio es una colección de objetos dentro del directorio que forman un subconjunto administrativo. Pueden existir diferentes dominios dentro de un bosque, cada uno de ellos con su propia colección de objetos y unidades organizativas.

Para poner nombre a los dominios se utiliza el protocolo DNS. Por este motivo, Active Directory necesita al menos un servidor DNS instalado en la red

3.1.3. Unidad Organizativa (UO)

En inglés Organizational Unit (OU). Es la unidad jerárquica inferior al dominio que puede tener objetos y/u otras UO. Más adelante las usaremos para la creación de GPOs.

Información

Las Unidades Organizativas (UO) son contenedores dentro del Active Directory

3.1.4. Objeto

La palabra Objeto se utiliza como nombre genérico para referirnos a cualquiera de los componentes que forman parte del directorio, como una impresora o una carpeta compartida, pero también un usuario, un grupo, etc. Incluso podemos utilizar la palabra objeto para referirnos a una Unidad organizativa.

Cada objeto dispondrá de una serie de características específicas (según la clase a la que pertenezca) y un nombre que permitirá identificarlo de forma precisa. En general, los objetos se organizan en tres categorías:

- **Usuarios**: identificados a través de un nombre (y, casi siempre, una contraseña), que pueden organizarse en grupos, para simplificar la administración.
- **Recursos**: que son los diferentes elementos a los que pueden acceder, o no, los usuarios según sus privilegios. Por ejemplo, carpetas compartidas, impresoras, etc.
- **Servicios**: que son las diferentes funciones a las que los usuarios pueden tener acceso. Por ejemplo, el correo electrónico.

3.1.5. Controlador de dominio

Un Controlador de Dominio (domain controller) contiene la base de datos de objetos del directorio para un determinado dominio, incluida la información relativa a la seguridad. Además, será responsable de la autenticación de objetos dentro de su ámbito de control.

En un dominio dado, puede haber varios controladores de dominio asociados, de modo que cada uno de ellos represente un rol diferente dentro del directorio. Sin embargo, a todos los efectos, todos los controladores de dominio, dentro del mismo dominio, tendrán la misma importancia.

3.1.6. Árbol

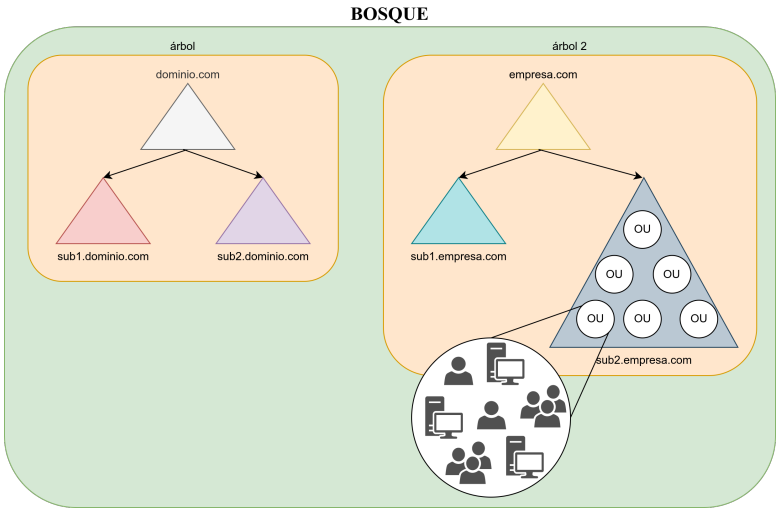
Un Árbol es simplemente una colección de dominios que dependen de una raíz común y se encuentran organizados como una determinada jerarquía. Dicha jerarquía también quedará representada por un espacio de nombres DNS común.

De esta forma, sabremos que los dominios **empresa.com** e **informatica.empresa.com** forman parte del mismo árbol, mientras que **empresa.com** y **linux.com** no.

Si un determinado usuario es creado dentro de un dominio, éste será reconocido automáticamente en todos los dominios que dependan jerárquicamente del dominio al que pertenece.

3.1.7. Bosque

El Bosque es el mayor contenedor lógico dentro de Active Directory, abarcando a todos los dominios dentro de su ámbito. Los dominios están interconectados por Relaciones de confianza transitivas que se construyen automáticamente (consultar más adelante el concepto de Relación de confianza). De esta forma, todos los dominios de un bosque confían automáticamente unos en otros y los diferentes árboles podrán compartir sus recursos.



Como ya hemos dicho, los dominios pueden estar organizados jerárquicamente en un árbol que comparte un espacio de nombres DNS común. A su vez, **diferentes árboles pueden estar integrados en un bosque**. Al tratarse de **árboles diferentes, no compartirán el mismo espacio de nombres**.

3.1.8. Relaciones de confianza

En el contexto de Active Directory, las Relaciones de confianza son un método de comunicación seguro entre dominios, árboles y bosques. Las relaciones de confianza permiten a los usuarios de un dominio del Directorio Activo autenticarse en otro dominio del directorio.

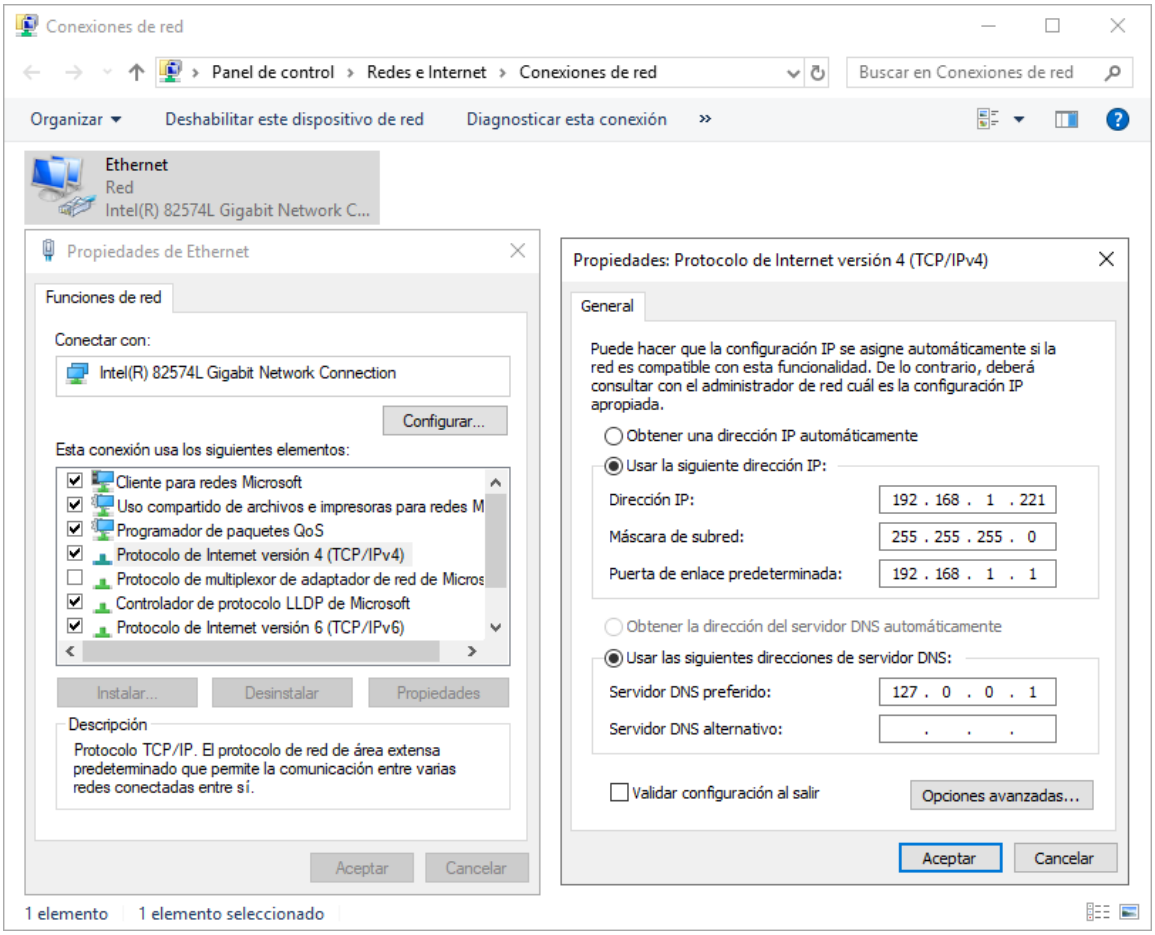
3.2. Configurando Windows Server como servidor en Red

Antes de promocionar el servidor a controlador de dominio, y por tanto convertirlo en un servidor en red, debemos realizar unos pasos previos de configuración. Los pasos que tendremos que realizar serán los

siguientes:

- Configurar una **IP estática** al equipo: Todos los servidores en una red (tanto pública como privada) debe de tener una IP estática en la misma. Deberemos de conocer el direccionamiento de la red, y confirmar que la IP que vamos a configurar no está siendo utilizada por otro equipo.

  - El cambio de IP se realiza en "**Configuración de Red e Internet**", en este caso "Protocolo de Internet versión 4" y poniendo la IP correspondiente a nuestra red.

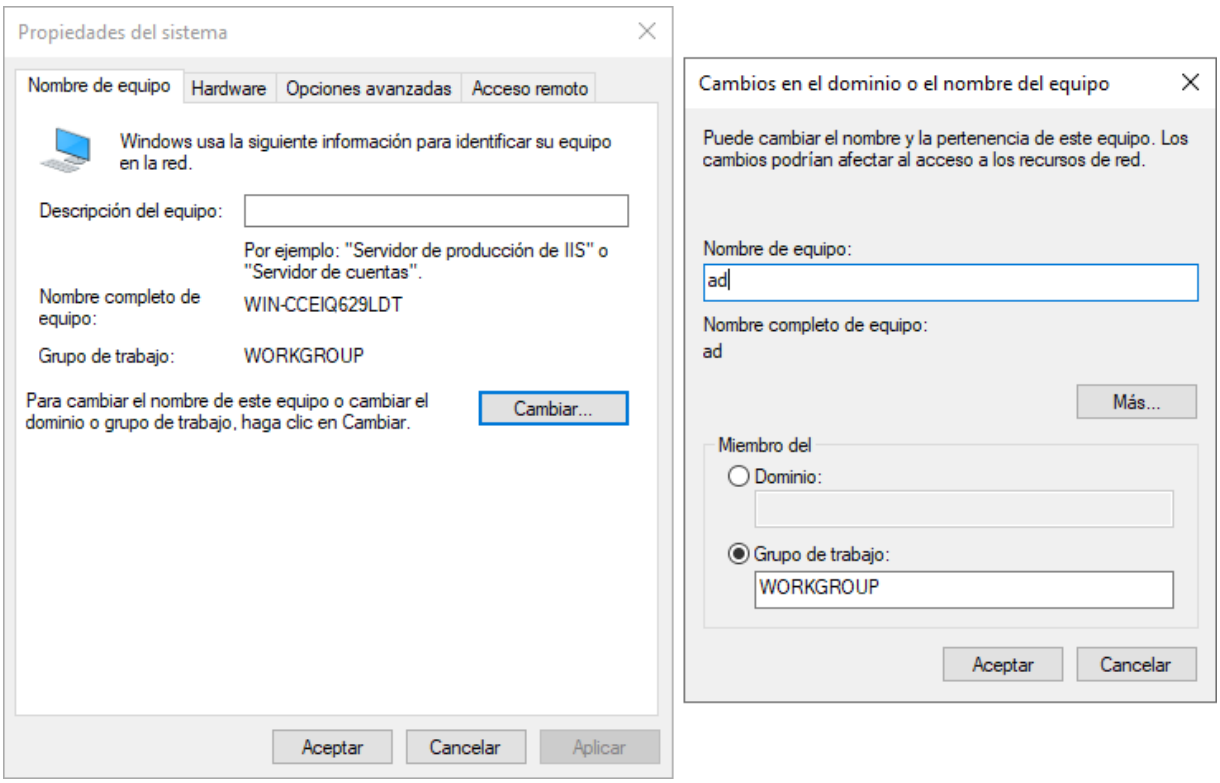


  - También vamos a configurar como **Servidor DNS preferido** la IP localhost, "127.0.0.1", para que posteriormente Active Directory ejerza de DNS local.

- Cambiar el **nombre del equipo**: Para que los servidores sean fácilmente identificables, debemos proporcionarles un nombre de equipo que indique cuáles son sus funciones. En nuestro caso, podemos llamarlo "**AD**" (de Active Directory) o "**DC**" (de *Domain Controller*). Tras realizar el cambio, el servidor deberá reiniciarse. El cambio se puede realizar desde varias partes de Windows Server, como por ejemplo:

  - Click derecho en Inicio → Sistema

  - Panel de control → Sistema y Seguridad → Sistema → Cambiar Configuración

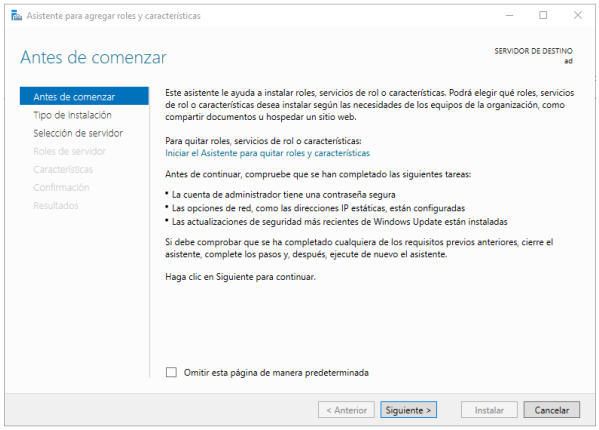


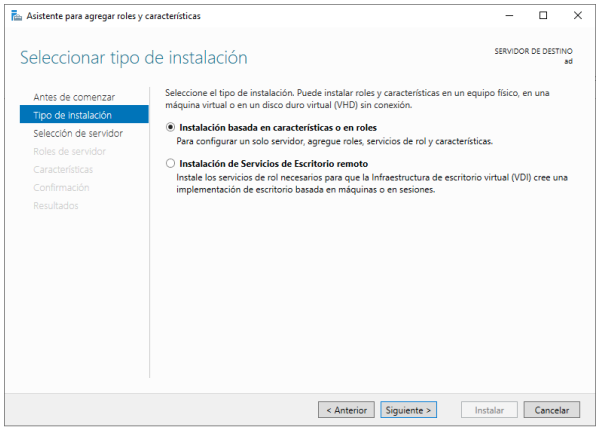
- Asegurar que la zona horaria es la adecuada
  - Comprobar servicio de hora
  - Comprobar que la actualización de la hora es automática

3.3. Instalar Active Directory

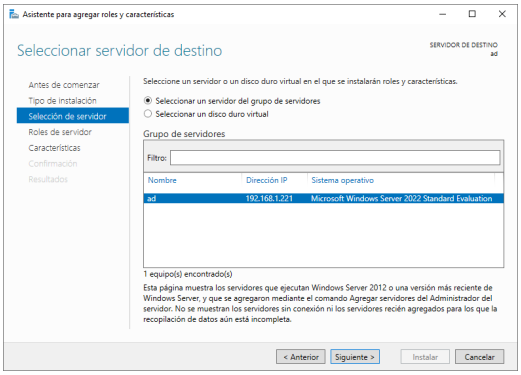
Para promocionar el servidor a controlador de dominio de Active Directory debemos de realizar la instalación del rol. Para ello, iremos al "**Administrador del Servidor**".

Haremos click en "Administrar", "Agregar roles y características" y nos saldrá una nueva ventana, con un aviso del asistente y al darle a "Siguiente", elegiremos la primera opción, "**Instalación basada en características y roles**" y volveremos a darle a "Siguiente" en el asistente:

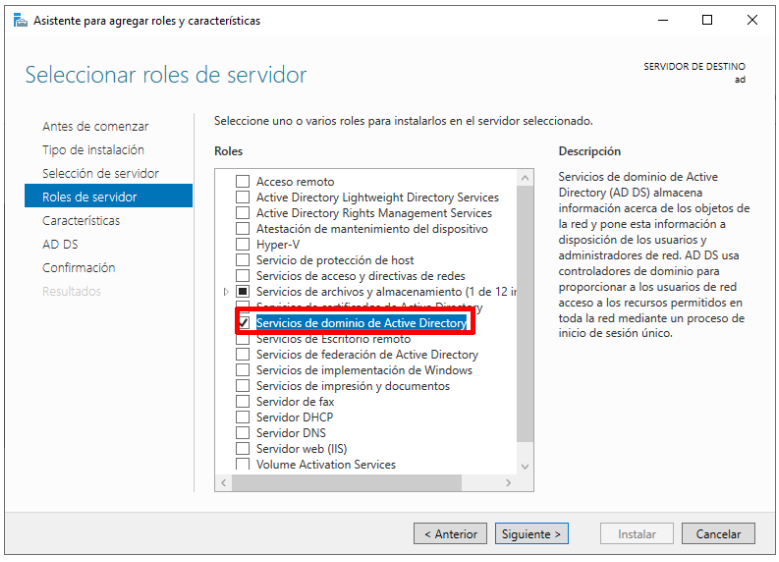




El siguiente paso será elegir el servidor dónde queremos realizar la instalación del rol de Active Directory. De primeras, puede parecer raro que nos pregunte en dónde queremos realizar la instalación, pero es fácil de entender cuando desde un servidor podremos llegar a controlar otros. De momento, sólo nos aparecerá el propio servidor donde estamos realizando la instalación, por lo que lo dejamos seleccionado y le damos a “Siguiente”.



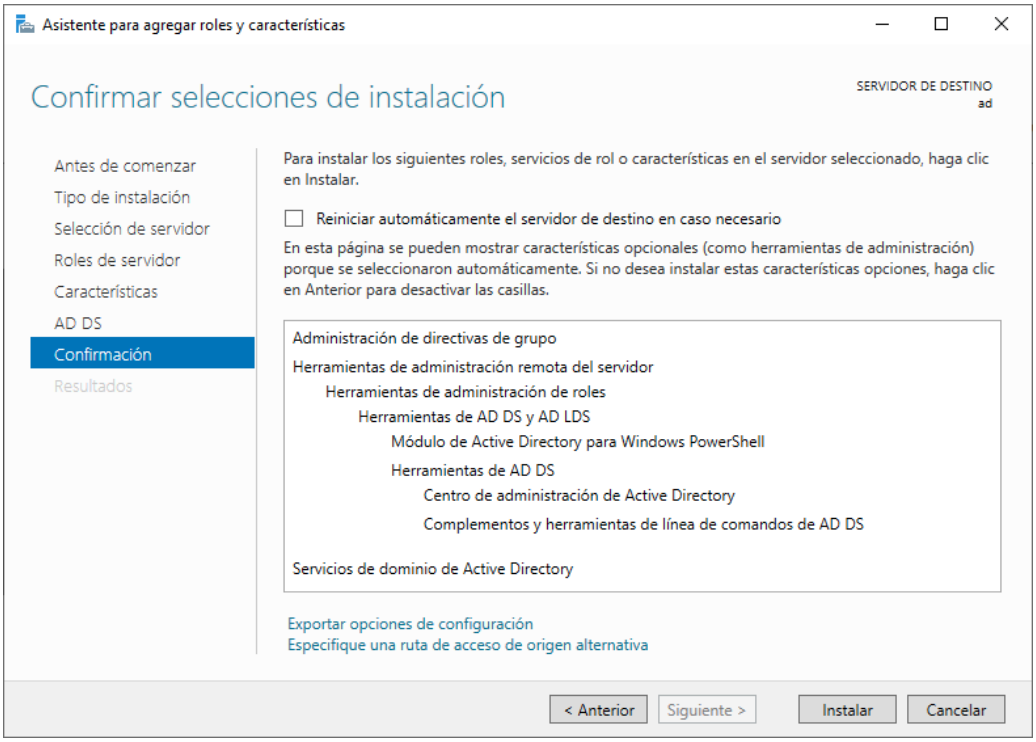
En el siguiente apartado podremos realizar la elección del rol que queremos añadir a nuestro servidor, y en este caso deberemos hacer click y asegurarnos que la opción está seleccionada en la opción “**Servicios de dominio de Active Directory**”.



Al hacer click en ella, nos saldrá una nueva ventana en la que nos avisa que se van a añadir herramientas necesarias para este rol, por lo que le daremos a “Agregar características” en esa nueva ventana. Le daremos a “Siguiente” para pasar al siguiente paso de la instalación.

El siguiente paso es el de “**Características**”, en la que aparecen distintas opciones con su descripción. De momento no vamos a realizar ninguna instalación en este apartado. Le daremos a Siguiente.

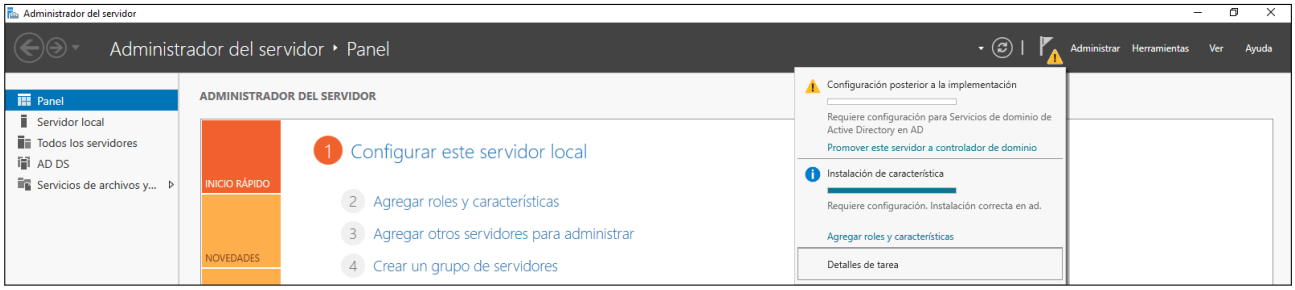
En el último paso nos aparecerá una confirmación de los pasos que se van a realizar y sólo nos quedará darle a "**Instalar**":



Tras esperar unos segundos, el proceso terminará y el botón de "Instalar" se cambiará por el de "Cerrar".

3.4. Configurar Active Directory

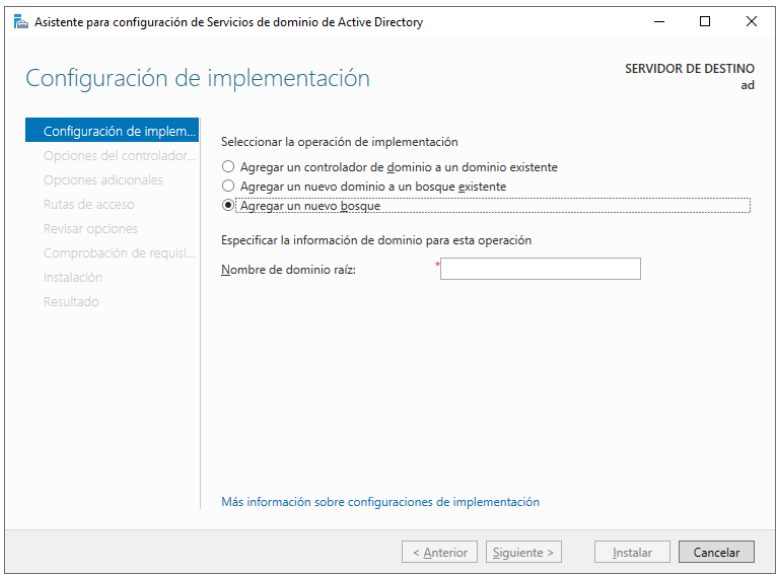
Tras realizar la instalación del rol de **Active Directory** en nuestro Windows Server, aparecerá un mensaje en la consola de administrador para que realicemos la configuración del mismo.



En el desplegable haremos click en "**Promover este servidor a controlador de dominio**", y nos abrirá una nueva ventana en la que tendremos que realizar una serie de pasos para realizar la configuración del Active Directory recién instalado.

3.4.1. Tipos de implementación

En el primer paso a la hora de configurar Active Directory aparecen tres opciones distintas que debemos de entender antes de proceder con la configuración. En la siguiente pantalla aparecen las tres opciones:



- **Agregar un controlador de dominio a un dominio existente**: lo utilizaremos cuando queramos tener Alta Disponibilidad en nuestra infraestructura, ya que el nuevo servidor actuará de réplica del servidor que actúe de controlador actualmente.

- **Agregar un dominio a un bosque existente**: cuando necesitemos añadir un nuevo dominio (para realizar diferenciación a otro que ya tengamos instalado) a un bosque ya existente.

- **Agregar un nuevo bosque**: Utilizado para realizar instalaciones nuevas creando un nuevo dominio.

3.4.2. Nombre de dominio raíz

A la hora de elegir un dominio para nuestro Active Directory existen varias opciones:

- Elegir un **TLD** (Top Level Domain) válido, registrado a tu empresa. Ejemplo: **miempresa.com**

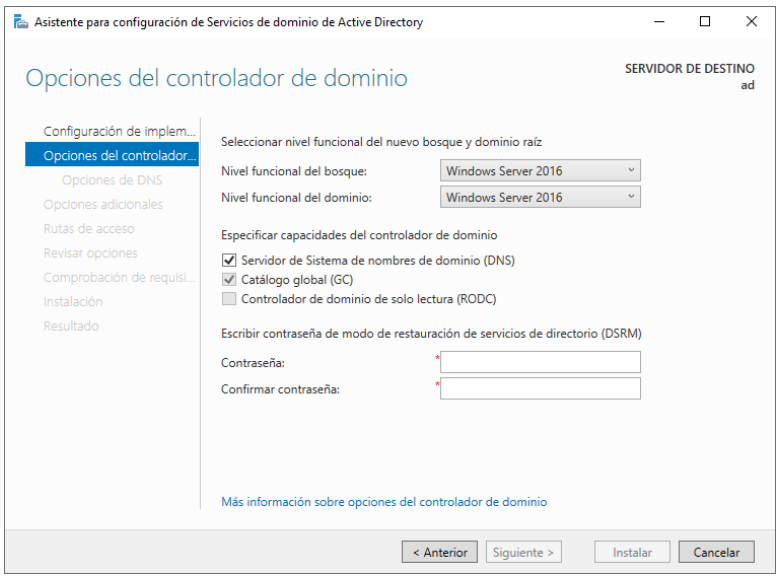
- Usar un **subdominio de un TLD** válido. Ejemplo: corp.miempresa.com Usar un **TLD no válido**. Ejemplo: miempresa.local, miempresa.corp, …

- En principio, y salvo que tengamos una razón para ello, elegiremos la primera opción y le daremos al botón de “Siguiente”.

Dado que estamos realizando una instalación nueva, utilizaremos la última opción.

3.4.3. Opciones del controlador de dominio

En el siguiente paso de configuración del Active Directory tendremos distintas opciones que debemos entender para proceder a su configuración:

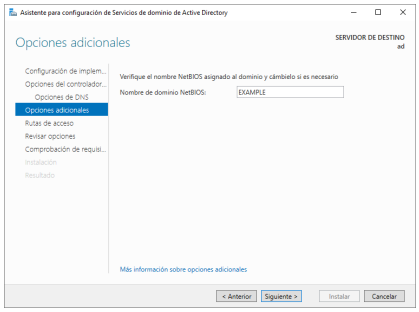


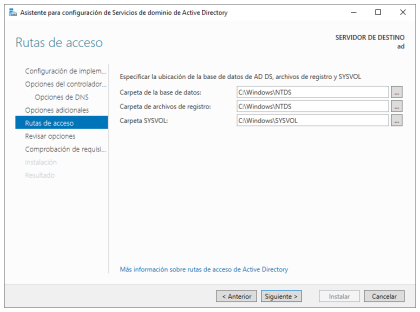
Los niveles funcionales determinan las funcionalidades disponibles de dominio o bosque de Active Directory, y determinan los sistemas operativos Windows Server que se pueden ejecutar en los controladores. Normalmente, **deberíamos elegir las últimas versiones disponibles** para poder utilizar el mayor número de características posibles. En caso de que necesitemos compatibilidad con versiones anteriores, entonces elegiremos la versión correspondiente.

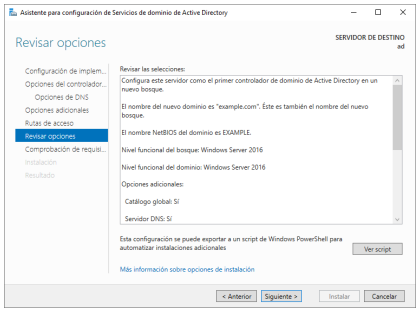
También nos va a pedir la contraseña de restauración de servicios de directorio DSRM. Con ella podremos iniciar sesión cuando no se esté ejecutando el servicio de Active Directory, ya sea porque se ha detenido o porque hemos tenido que iniciar en modo DSRM el servidor.

3.4.4. Otras opciones

Una vez realizado el paso anterior, y tras darle a “Siguiente”, nos aparecerán nuevos pasos que podremos pasarlos sin realizar modificaciones, ya que las configuraciones por defecto se corresponden al dominio que hayamos introducido o rutas donde se almacena la información. El último paso será para revisar y realizar la instalación:







Tras estos pasos revisaremos las opciones y le daremos a “Instalar”.

4. Añadir un equipo Windows 10 al dominio

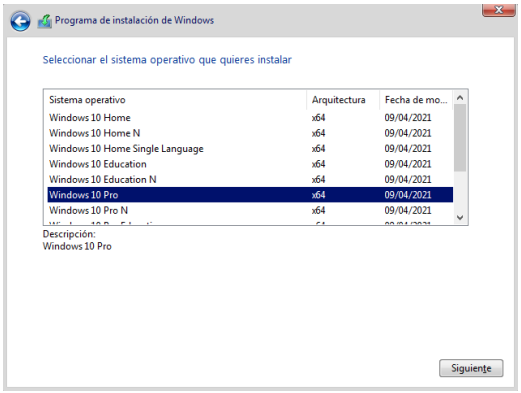
Tras seguir los pasos previos, podremos añadir equipos al dominio creado en Active Directory y hacer uso de las configuraciones, usuarios y restricciones que iremos creando en apartados sucesivos. Para que un

equipo Windows pueda pertenecer a un Active Directory debe ser de la familia "**Pro**", como por ejemplo: Windows 7 Pro, Windows 10 Pro…

4.1. Instalación de Windows 10 Pro

Para realizar la instalación de un equipo con Windows 10, podremos hacer uso de la ISO que podremos descargar desde la página de Microsoft.

No se van a explicar todos los pasos, ya que el sistema de instalación es sencillo. El único apartado donde deberemos fijarnos es en la pantalla de la derecha.



Tal como se ve en la imagen, elegiremos la opción "Windows 10 Pro", y continuaremos.

Al finalizar la instalación, el equipo se reiniciará y podremos introducir el usuario y contraseña que hemos creado durante la instalación.

Información

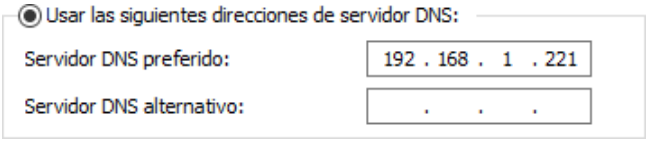
El usuario creado en la instalación es una cuenta LOCAL del equipo

4.2. Pasos previos

El equipo Windows 10 Pro deberá estar situado en la misma red del Windows Server, o permitir todas las conexiones que sean necesarias de estar en otra red.

Es recomendable modificar el nombre del equipo para que posteriormente sea más fácil de encontrarlo dentro del Active Directory, ya que el nombre por defecto de instalación es aleatorio. Para realizar el cambio, podremos seguir los pasos indicados previamente para Windows Server, y en este caso elegiremos el nombre "win10".

Y por último, en la configuración de red del equipo Windows 10 configuraremos el servidor DNS preferido para que sea la IP del Windows Server 2022. Este cambio se podría realizar de manera automática si modificamos la configuración del DHCP de la red.

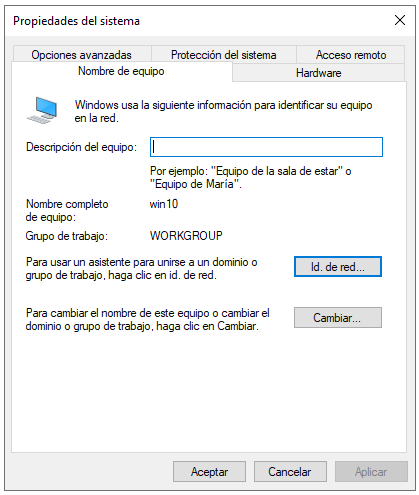


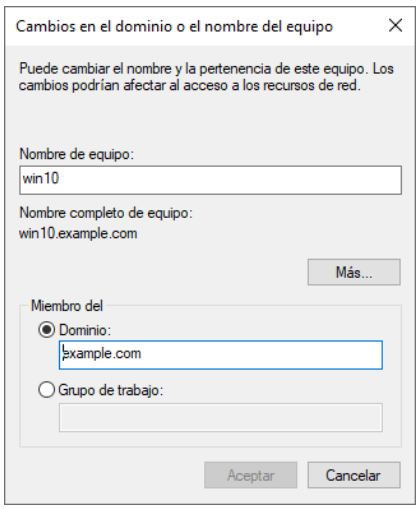
4.3. Añadir el equipo al dominio

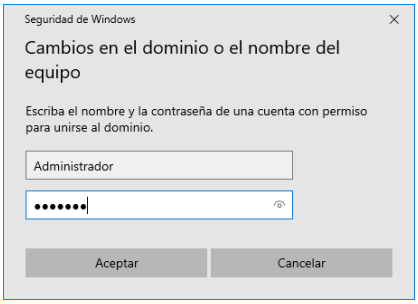
Tras realizar los pasos previos, para añadir un equipo a un dominio de Active Directory tendremos que ir a la **Propiedades del Sistema** de Windows (dependiendo de la versión de Windows se puede llegar a esta

pantalla de varias maneras).

Le daremos a “Cambiar”. Si no hemos cambiado el nombre al equipo, podremos hacerlo en la nueva ventana, y a la par seleccionaremos el “Dominio” que hemos creado previamente, le daremos a “Aceptar” y nos aparecerá una nueva ventana donde **introduciremos el usuario “Administrador” y contraseña del Windows Server 2022**.







El equipo se conectará al servidor Windows Server 2022, comprobará que los datos son correctos, añadirá el equipo al Active Directory y nos pedirá reiniciar.

4.4. Diferenciar usuario local y usuario de Active Directory

Una vez que el equipo Windows 10 se ha añadido al Active Directory, nos vamos a poder loguear a él de dos maneras distintas:

- **Usuario local**: Tras la instalación de Windows 10 sólo existe un usuario, que es el que hemos creado durante la instalación y es **Administrador local**. Este usuario no deberíamos utilizarlo salvo que el equipo tuviese algún problema que no se pudiese solventar desde Windows Server, quizá porque se haya salido del dominio, no tiene conexión a la red, …
- **Usuario de Active Directory**: La gestión de usuarios va a ser centralizada en Active Directory tal como vamos a ver en el apartado de gestión de grupos y usuarios, por lo que cualquier usuario que sea creado en Active Directory ahora mismo podría hacer login en este equipo.

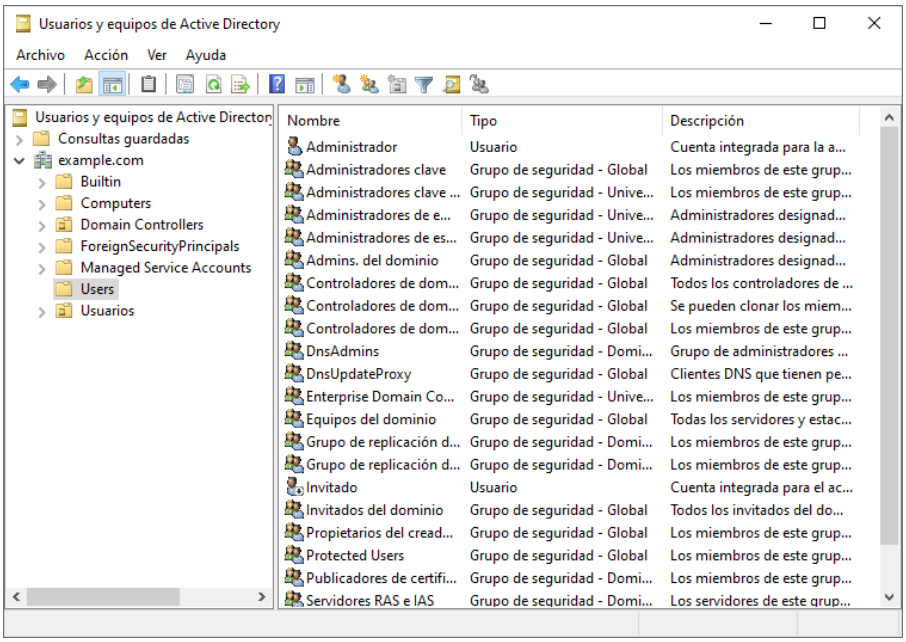
A partir de ahora, cuando nos vuelva a salir la pantalla para introducir el usuario y contraseña, nos aparecerá a la izquierda el usuario que hayamos creado durante la instalación de Windows 10 y la opción “**Otro Usuario**”. Al seleccionar esta opción, tendremos que introducir un usuario y la contraseña de un usuario **que exista en Active Directory**.

Tal como se puede ver, ya nos viene marcado que al introducir el usuario se va a iniciar la sesión en el dominio creado anteriormente. También podremos indicar lo siguiente como nombre de usuario:

- **usuario@example.com**: cuando necesitemos iniciar sesión con "usuario" del dominio "example.com".
- **ruben@dominio.com**: cuando queramos usar como inicio de sesión el usuario "ruben" del dominio "dominio.com".
- **.\usuario**: donde el punto hace referencia a iniciar sesión local y "usuario" es el usuario administrador local que hemos creado durante la instalación.

5. Gestión de grupos y usuarios

Para crear usuario y grupos, dentro del "**Administrador del Servidor**", en el desplegable "**Herramientas**", elegiremos la opción "**Usuarios y equipos de Active Directory**", y nos aparecerá la siguiente ventana:



Tal como se puede observar, a la izquierda tenemos un desplegable del dominio que hemos creado durante la instalación de Active Directory, y al seleccionar "Users" nos aparece a la derecha los **usuarios** y **grupos de seguridad** que el sistema ha creado por defecto.

5.1. Grupos dentro del Active Directory

Los grupos dentro del Active Directory son objetos que **pueden contener usuarios, contactos, equipos u otros grupos**.

¡Atención!

No hay que confundir grupos con las Unidades Organizativas.

Al agregar un objeto a un grupo, ese objeto recibe todos los derechos asignados al grupo y todos los permisos asignados al grupo para todos los recursos compartidos.

Podemos utilizar los grupos para simplificar algunas tareas, como:

- **Simplificar la administración**: Podemos asignar permisos al grupo y éstos afectarán a todos sus miembros.
- Delegar la **administración**: Podemos utilizar la directiva de grupo para asignar derechos de usuario una sola vez y, más tarde, agregar los usuarios a los que queramos delegar esos derechos.
- **Crear listas de distribución de correo electrónico**: Sólo se utilizan con los grupos de distribución que comentaremos más abajo.

Para crear grupos debemos tener clara la infraestructura de la empresa, los usuarios que tenemos y los grupos a los que va a pertenecer cada usuario. Esto es importante ya que la organización de los usuarios es una labor tediosa que en caso de tener que rehacerse se pierde tiempo en realizarlo.

Por otro lado, hay que recordar que cuando creemos el grupo el nombre será único dentro del dominio.

5.1.1. Tipos de grupos

Windows Server admite dos tipos de grupos:

- **Distribución**: Se usa para crear listas de distribución de correo electrónico. Sólo se pueden usar con aplicaciones de correo electrónico (como Exchange Server) para enviar correo electrónico a colecciones de usuarios. Los grupos de distribución no tienen seguridad habilitada.
- **Seguridad**: Se usan para proporcionar de manera eficaz la asignación del acceso a los recursos de la red. Con los grupos de seguridad se pueden asignar derechos de usuario a grupos de seguridad en Active Directory permisos a grupos de seguridad de recursos.

5.1.2. Ámbito de grupos

Los grupos se caracterizan por un **ámbito** que identifica el grado en el que se aplica el grupo en el árbol de dominios o en el bosque. **El ámbito del grupo define dónde se pueden conceder permisos al grupo**. Los siguientes tres ámbitos de grupo se definen mediante Active Directory:

- **Dominio Local**: Sólamente se puede otorgar permisos sobre los recursos que se sitúan en el dominio donde está ubicado el grupo de dominio local.
- **Global**: Puede otorgar permisos sobre los recursos de cualquier dominio del bosque, o dominios de confianza.
- **Universal**: Puede otorgar permisos sobre cualquier dominio del mismo bosque o de bosques de confianza.

5.1.3. Grupos de seguridad predeterminados

Los grupos predeterminados, como el grupo administradores del dominio, son grupos de seguridad que se crean automáticamente al crear un dominio en Active Directory. Se pueden utilizar estos grupos predefinidos para controlar el acceso a los recursos compartidos y para delegar roles administrativos específicos para todo el dominio.

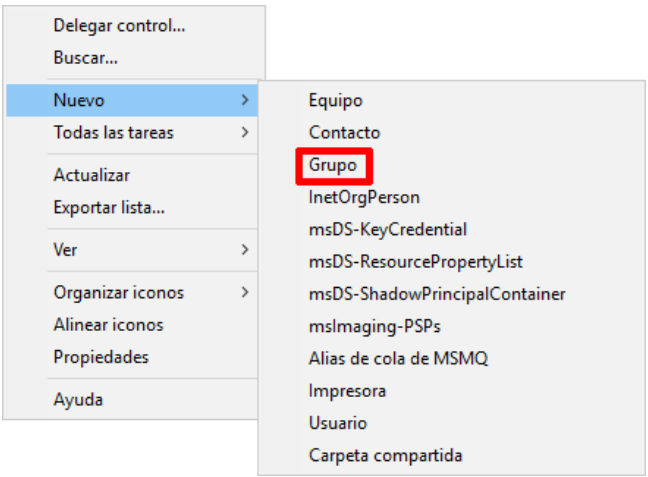
A muchos grupos predeterminados se les asigna automáticamente un conjunto de derechos de usuario que autorizan a los miembros del grupo a realizar acciones específicas en un dominio, cómo iniciar sesión en un sistema local o realizar copias de seguridad de archivos y carpetas. Por ejemplo, un miembro del grupo operadores de copia de seguridad tiene el derecho de realizar operaciones de copia de seguridad para todos los controladores de dominio del dominio.

Los grupos predeterminados se encuentran en el contenedor **Builtin** y en el contenedor usuarios de usuarios y equipos de Active Directory.

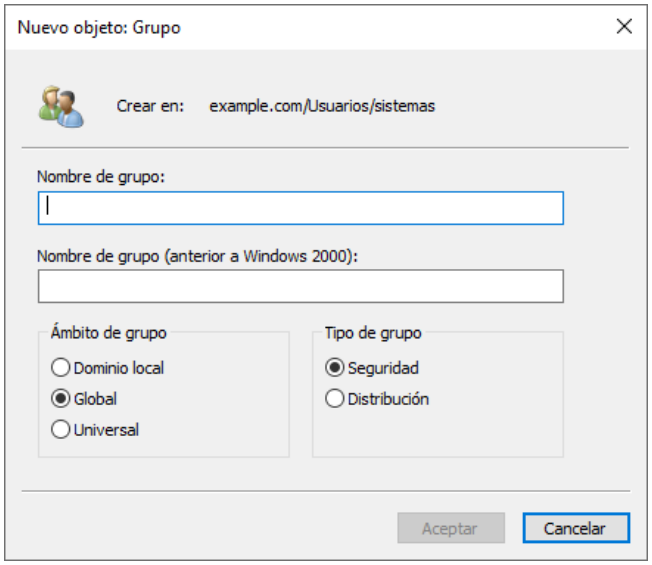
5.1.4. Crear un grupo para usuarios

Windows Server ya trae una serie de grupos creados por defecto y cada uno de ellos cuenta con sus características.

Para crear un grupo propio tendremos que ir a "**Usuarios y equipos de Active Directory**", seleccionar **Users** y hacer click derecho para elegir "**Nuevo**":



Al darle a la opción "Grupo" tal como aparece en el menú desplegable, nos aparecerá una nueva ventana donde podremos elegir el nombre del grupo y las opciones comentadas previamente: qué "Ámbito de grupo" queremos que sea y el "Tipo de grupo":



Tras poner las opciones que mejor nos convenga, nos aparecerá el nuevo grupo, al que posteriormente podremos añadir usuarios.

5.2. Usuarios

En Windows Server, una cuenta de usuario es un objeto que posibilita el acceso a los recursos del dominio de dos modos diferentes:

- Permite **autenticar la identidad de un usuario**, porque sólo podrán iniciar una sesión aquellos usuarios que dispongan de una cuenta en el sistema asociada a una determinada contraseña.
- Permite **autorizar**, o **denegar**, el acceso a los recursos del dominio, porque, una vez que el usuario haya iniciado su sesión sólo tendrá acceso a los recursos para los que haya recibido los permisos correspondientes.

Cada cuenta de usuario dispone de un **identificador** de seguridad (**SID** o Security IDentifier) **que es único en el dominio**.

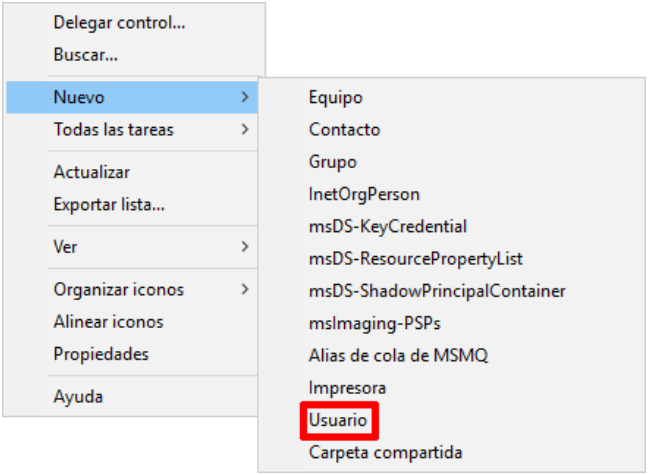
5.2.1. Usuarios predeterminados

Cuando se crea el dominio, se crean también dos nuevas cuentas:

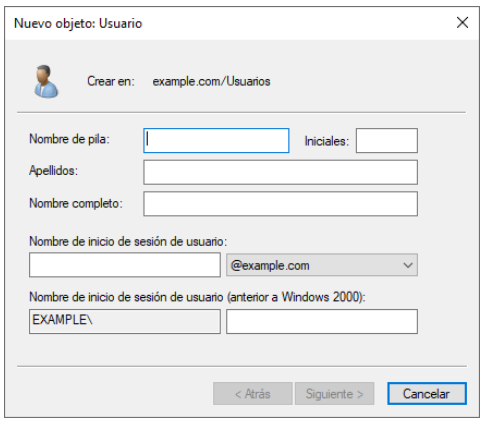
- **Administrador**: Tiene control total sobre el dominio y no se podrá eliminar ni retirar del grupo Administradores (aunque sí podemos cambiarle el nombre o deshabilitarla).
- **Invitado**: Está deshabilitada de forma predeterminada y, aunque no se recomienda, puede habilitarse, por ejemplo, para permitir el acceso a los usuarios que aún no tienen cuenta en el sistema o que la tienen deshabilitada. De forma predeterminada no requiere contraseña, aunque esta característica, como cualquier otra, puede ser modificada por el administrador.

5.2.2. Agregar cuenta de usuario no-administrador

Para crear un nuevo usuario, por ejemplo **user1**, tendremos que elegir dónde lo queremos guardar. Haremos igual que con el grupo, click derecho y elegir la opción “Usuario” en este caso:



Al seleccionar la opción de “Usuario”, nos aparecerá una nueva ventana donde podremos rellenar:



- **Nombre de pila**: Nombre del usuario. Este nombre se puede repetir en el servidor.
- **Iniciales**: Formato abreviado del nombre y apellidos
- **Apellidos**: Los apellidos del usuario
- **Nombre completo**: Se autocompleta con lo rellenado en el nombre y los apellidos
- **Nombre de inicio de sesión de usuario**: El nombre que el usuario deberá introducir al encender el ordenador y aparecer la pantalla de inicio de sesión.

¡Atención!

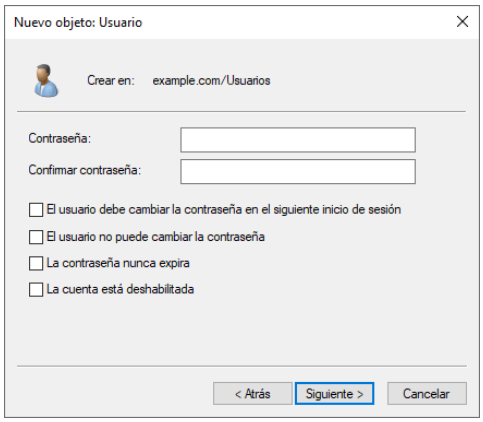
El nombre de inicio de sesión no debe repetirse.

- **Nombre de inicio de sesión de usuario (Anterior a Windows 2000)**: Debería coincidir con el nombre de inicio de sesión.

Tras darle a “Siguiente” nos aparecerá la pantalla donde debemos introducir:

- **Contraseña**: La contraseña que el usuario deberá introducir al iniciar sesión.
- **Confirmar contraseña**: Debe coincidir.

Y aparecen cuatro opciones que podemos marcar dependiendo de lo que necesitemos:



- Si permitimos que el usuario pueda cambiar la contraseña la siguiente vez que inicie sesión (al estar creando el usuario, será en el primer inicio de sesión).
- NO permitir que el usuario pueda cambiar la contraseña. Por lo tanto, en caso de querer modificarla se deberá realizar desde el servidor.
- Que no expire la contraseña, ya que por defecto caducan pasados 42 días.
- Deshabilitar la cuenta.

Al darle a “Siguiente” nos aparecerá el último paso que será el de confirmar la creación del usuario.

Información

Todas estas opciones pueden ser modificadas una vez creado el usuario.

5.3. Cuentas de equipo

Una cuenta de equipo sirve para autenticar a los diferentes equipos que se conectan al dominio, permitiendo o denegando su acceso a los diferentes recursos del dominio.

Del mismo modo que con las cuentas de usuario, las cuentas de equipo deben ser únicas en el dominio. Aunque una cuenta de equipo se puede crear de forma manual, lo habitual es que se crean en el momento en el que el equipo se une al dominio.

6. GPO: Directivas de grupos

Las GPO (*Group Policy Objets*) en castellano conocidas como "Objetos de Políticas de Grupo" (o simplemente directivas de grupo), son los objetos que incluye una serie de directivas (o políticas) que pueden aplicarse de manera centralizada a equipos y usuarios. Cada GPO establece una configuración del objeto al que afecta. Por ejemplo:

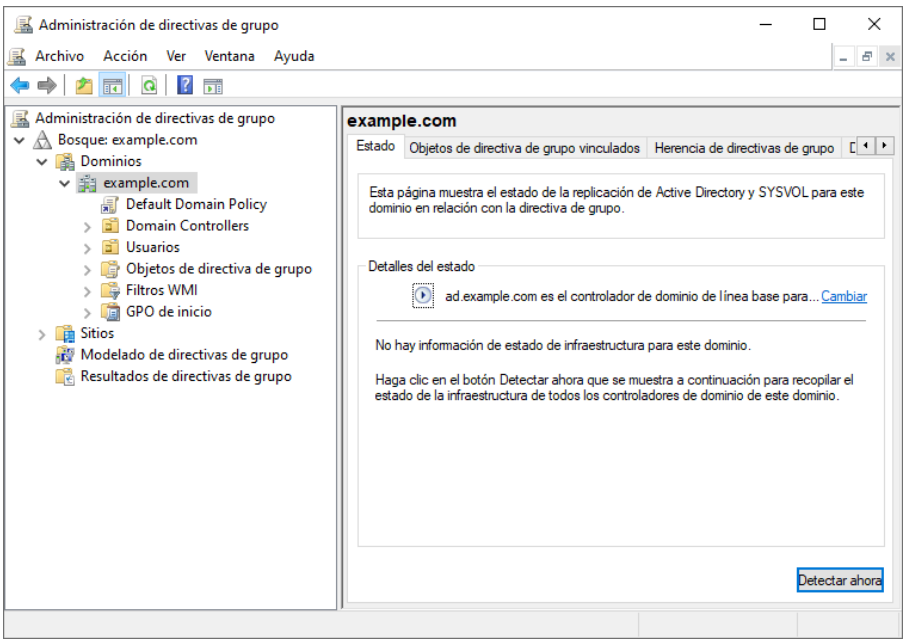
- Montar una unidad de red correspondiente
- Modificar configuración del firewall
- Ocultar el panel de control
- …

Información

Las Directivas de Grupos controlan lo que los usuarios y los equipos pueden y no pueden hacer.

Las GPO se vinculan a nuestra estructura de árbol al nivel al que deseamos configurar. Si vinculamos una GPO en el nombre de nuestro árbol, esta GPO afectará a todo el árbol. Si por el contrario vinculamos nuestra GPO en un nombre de dominio, esa GPO solo afectará a dicho dominio. **Las GPO se pueden vincular en sitios, dominios y unidades organizativas**. Tal como se puede ver, y pese a su nombre, las GPO **no se pueden asociar a un grupo de usuarios**.

Para crear una GPO, tendremos que ir al "Administrador del Servidor" y en "Herramientas" elegir "Administración de directivas de grupo" (también conocido como **GPMC.exe**), lo que nos abrirá la siguiente ventana:



Las GPOs no se aplican de manera inmediata, ya que el equipo debe reiniciarse o el usuario cerrar sesión. Podemos hacer que desde el equipo un usuario ejecute la orden "**gpupdate.exe**" y esto forzará la actualización de las GPO, pero requiere de la intervención del usuario. Más adelante vamos a ver cómo forzar la actualización de las GPO de manera remota desde el servidor.

6.1. Nivel de aplicación

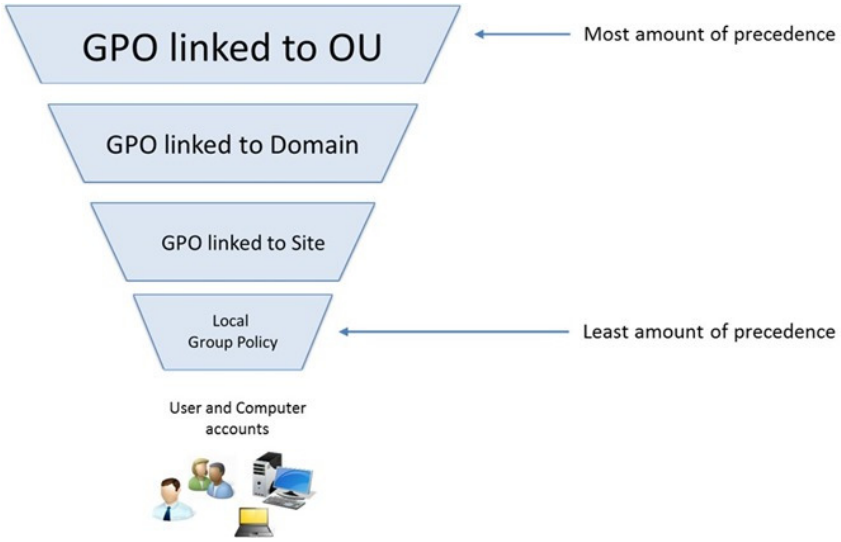
Dado que en un bosque podemos tener distintas GPOs que puedan afectar a un mismo objeto, debemos conocer que las directivas se aplican en este orden:

1. **Objeto de directiva de grupo local**: cada equipo tiene exactamente un objeto de directiva de grupo almacenado de forma local. Este objeto controla el procesamiento de las directivas de grupo de equipo y usuario.

2. **Sitio**: todos los GPO vinculados al sitio al que pertenece el equipo se procesan a continuación. El procesamiento se efectúa en el orden especificado por el administrador en la ficha **Objetos de directivas de grupo vinculados** del sitio en la Consola de administración de directivas de grupo (GPMC, Group Policy Management Console). La GPO con el **orden de vínculos** más bajo es la última en procesarse y, por tanto, tiene la máxima prioridad.

3. **Dominio**: el procesamiento de varias GPO vinculadas a un dominio se efectúa en el orden especificado por el administrador, en la ficha **Objetos de directivas de grupo vinculados** del dominio en GPMC. La GPO con el **orden de vínculos** más bajo es la última en procesarse y, por tanto, tiene la máxima prioridad.

4. **Unidades organizativas**: los GPO vinculados a la unidad organizativa que se encuentra en el nivel más alto de la jerarquía de Active Directory se procesan primero, luego las GPO vinculadas a su unidad organizativa secundaria y así sucesivamente. Por último, se procesan los GPO vinculados a la unidad organizativa que contiene el usuario o el equipo.



La Directiva de Grupo se aplica de manera jerárquica desde el grupo menos restrictivo (Sitio) al grupo más restrictivo (Unidad Organizativa). La Directiva de Grupo también es acumulativa.

El comportamiento respecto a la herencia y prioridad entre GPO en contenedores anidados puede ser refinado mediante los siguientes dos parámetros de configuración:

- **Exigido** (enforced). Este parámetro puede activarse independientemente a cada vínculo de un GPO. Si un vínculo de un GPO a un contenedor tiene este parámetro activado, sus políticas no pueden ser sobrescritas por GPO que se apliquen posteriormente.

- **Bloquear herencia** (de directivas) (Block Policy inheritance). Este parámetro pertenece a los contenedores del Directorio Activo. En particular, si un contenedor tiene este parámetro activado, se desactiva la herencia de las políticas establecidas en contenedores superiores, excepto aquellas que corresponden a GPO vinculados con el parámetro Exigido.

Para poder crear o modificar directivas de grupo deberemos ir al asistente de “Administración del Servidor → Herramientas → Administración de directivas de grupo” o desde el comando **GPMC.exe**.

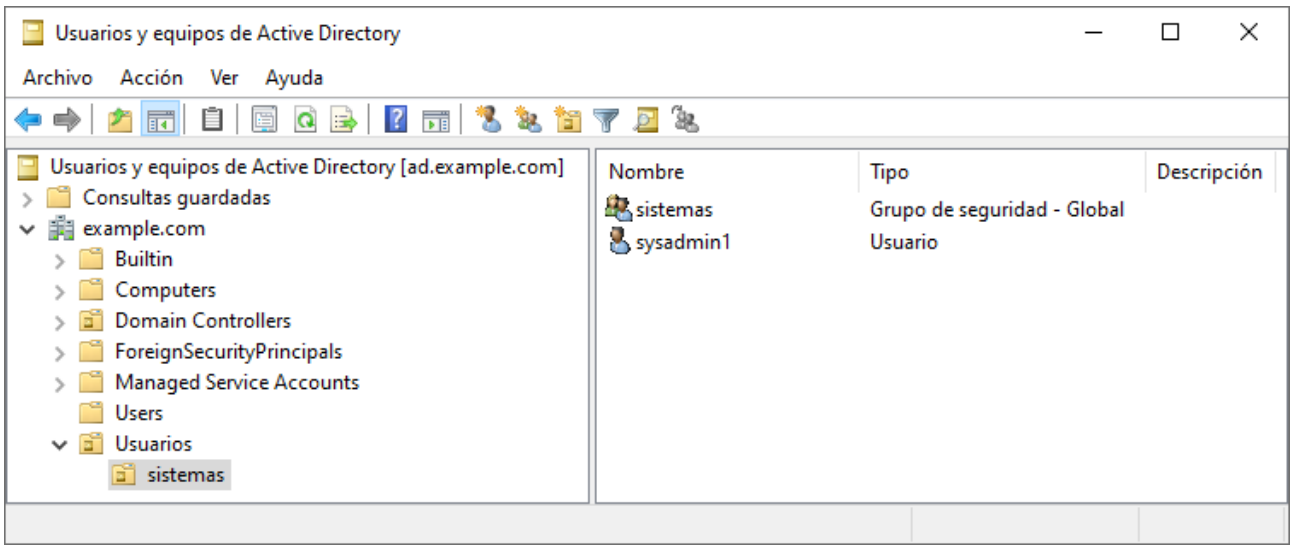
6.2. Ejemplos prácticos

A continuación se van a detallar la creación de distintas GPO que podremos utilizar de base para crear otras.

6.2.1. Añadir una unidad de red a usuarios

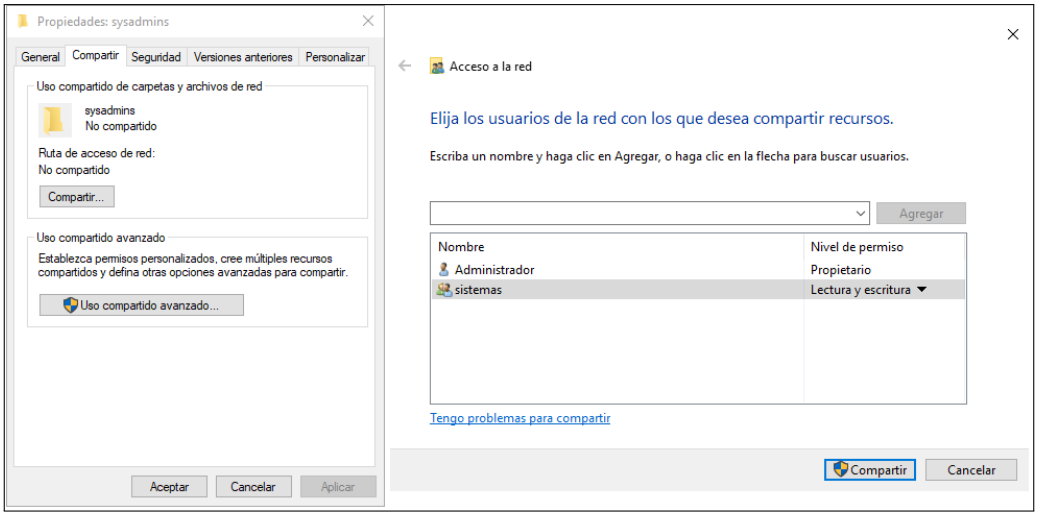
Es habitual que en una red los usuarios tengan que conectarse a unidades de red. Los usuarios finales no tienen por qué saber cómo conectarse a una unidad de red, y por tanto si les facilitamos el trabajo será más fácil que puedan acceder. En una red con Active Directory podemos indicar que los usuarios tengan distintas unidades de red conectadas en el equipo y con una letra correspondiente en el momento en el que se loguean.

Para el ejemplo se ha creado una Unidad Organizativa llamada “Usuarios” y dentro otra llamada “Sistemas” y dentro tenemos un usuario llamado “sysadmin1” que hemos añadido al grupo “sistemas”:



6.2.1.1. Crear una carpeta compartida

Se ha creado una carpeta que hemos dejado en el escritorio del usuario administrador, aunque lo ideal es ponerlo en una ruta más acorde, pero como prueba nos sirve. Esta carpeta va a ser compartida con el grupo "sistemas" que hemos creado previamente:



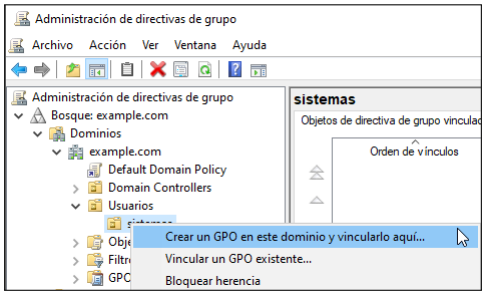
Tras realizar estos pasos, un usuario que pertenezca al grupo "sistemas" podría conectarse a la carpeta compartida, pero debería conocer la ruta. Vamos a automatizar esto a través de la GPO.

6.2.1.2. Crear GPO

Tal como hemos comentado previamente, la GPO va a permitir que cuando el usuario inicie sesión le aparezca la carpeta compartida como una unidad nueva con la letra que especifiquemos. Los pasos para crear la GPO son:

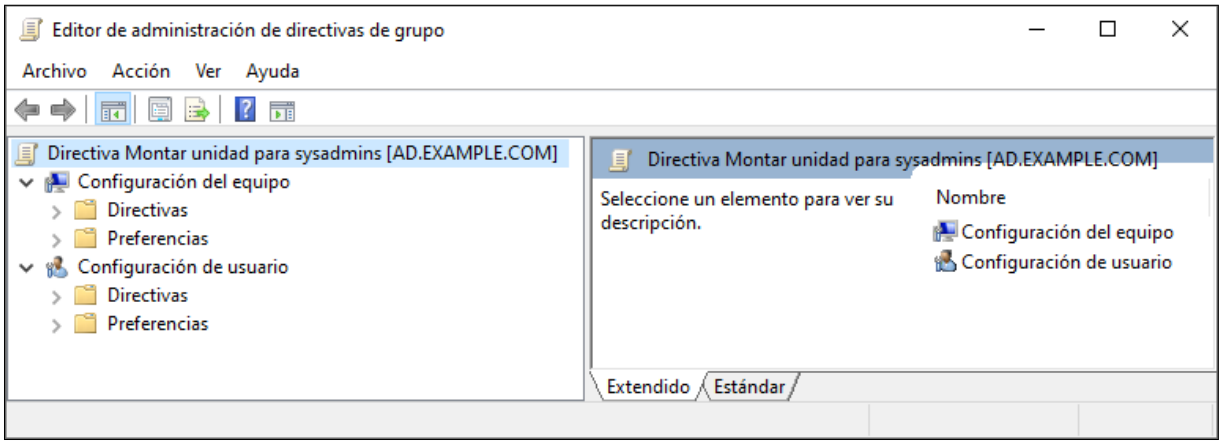
1. Ir a "Administración de directivas de grupo"

2. Crear GPO en la unidad organizativa en la que queremos que esté asociado.



3. Nos aparecerá una nueva ventana en la que podremos introducir un nombre para la GPO que vamos a crear. Este nombre debería darnos información de para qué sirve la GPO, por lo tanto, hay que elegir un nombre representativo, por ejemplo: “montar unidad para sysadmins”

4. Una vez creada, debemos configurar qué acciones queremos que realice. Para ello, haremos click derecho encima de la GPO y le daremos a “Editar”, lo que nos abrirá una nueva ventana con el programa de edición de directivas de grupo, con la directiva que hemos seleccionado.



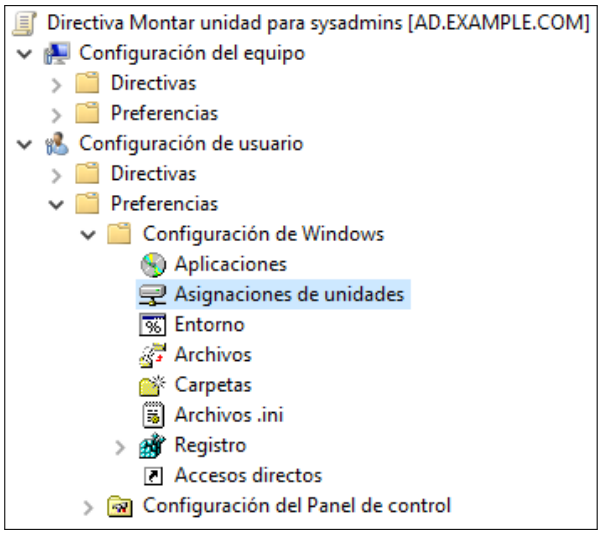
Tal como se puede ver en la imagen, arriba a la izquierda aparece el nombre de la GPO y el servidor en el que está creada. Debajo aparece si va a ser de tipo Configuración del equipo o configuración para usuario.

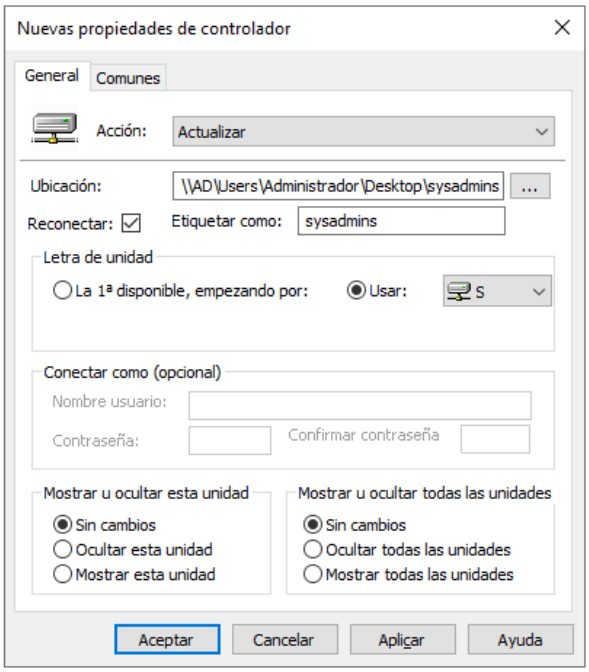
5. Vamos a crear una configuración para usuarios, por lo tanto iremos a “*Configuraciones de usuario → Preferencias → Configuración de Windows → Asignaciones de unidades”. Le damos a ”Nueva → Unidad asignada*” y deberemos rellenar los datos indicando:

   a. Ubicación de los datos

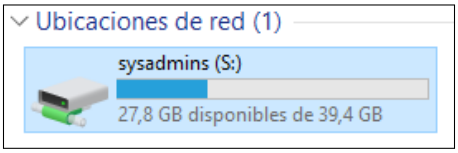
   b. Etiqueta

   c. Letra de unidad

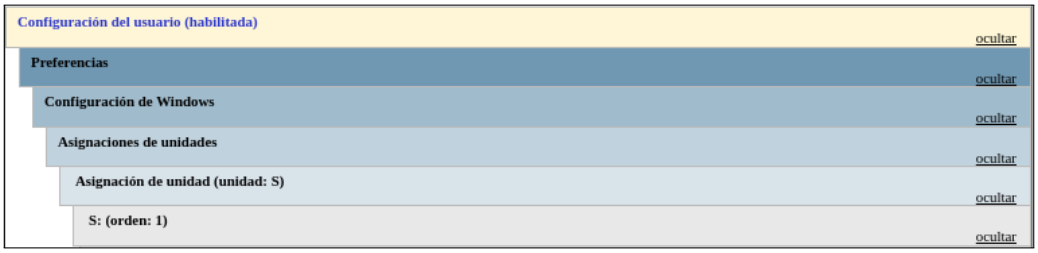




Una vez hecho esto, cuando cualquier usuario que esté creado en la Unidad Organizativa “sistemas” inicie sesión, tendrá una unidad compartida en el apartado “Este equipo”:



De esta manera, realizando una única configuración será válida para todos los usuarios que tengamos en esa Unidad Organizativa. El resumen de la GPO es:

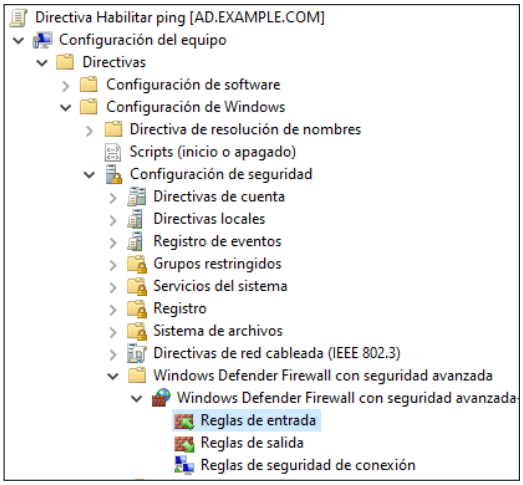


6.2.2. Permitir conexiones ping

Es habitual que los equipos Windows 10 no permitan realizar conexiones del protocolo ICMP (ping) ya que el firewall las rechaza. Por otro lado, para tener controlado los equipos que están encendidos en la red es interesante que esté habilitado.

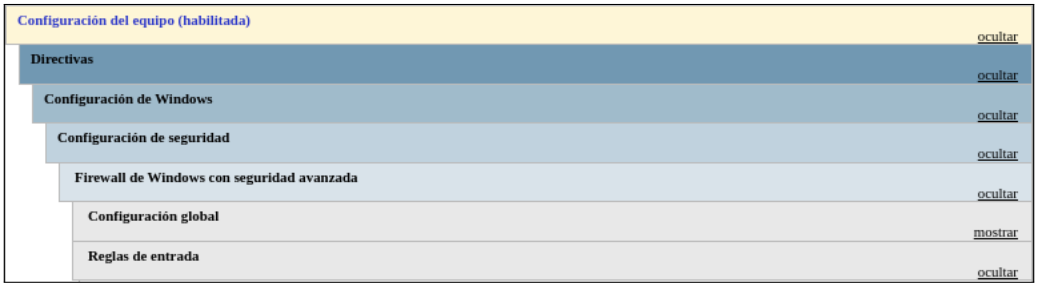
Dado que ir configurando equipo a equipo el firewall es una tarea tediosa, el realizar la configuración de manera automática mediante una GPO es sencillo y nos permite ver cómo podemos realizar configuraciones que en este caso afectan al equipo y no al usuario.

En este caso vamos a crear una GPO que afecte a todos los equipos del dominio, por lo tanto haremos click derecho sobre el dominio y crearemos la GPO en él, poniéndole un nombre como "**Habilitar ping**".



Editamos la GPO y esta vez tenemos que desplegar los menús hasta llegar a: "*Configuración del equipo → Directivas → Configuración de Windows → Configuración de Seguridad → Windows Defender Firewall con seguridad avanzada → Windows Defender Firewall con seguridad avanzada →* ***Reglas de entrada***".

Al llegar a este punto, haremos click derecho y le daremos a "Nueva regla" y vamos a seguir el asistente de creación de una nueva regla personalizada que permita el protocolo ICMPv4 de entrada desde cualquier equipo.



Una vez creada la GPO se aplicará cuando los equipos reinicien. Para evitar tener que reiniciar al crear nuevas GPO, vamos a crear la siguiente GPO que permitirá la administración del equipo de manera remota.

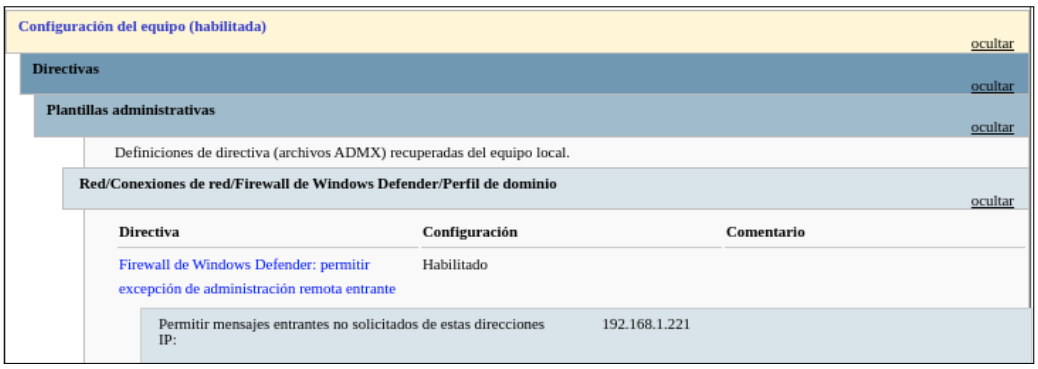
6.2.3. Administración remota de equipos

Tal como hemos dicho anteriormente, las GPOs no se aplican de manera inmediata, y dependiendo de si son de equipo o de usuario se deberá reiniciar o cerrar sesión.

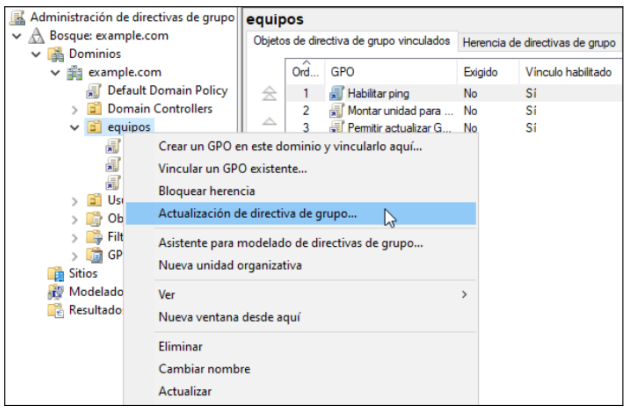
Para poder actualizar las GPOs de manera remota vamos a crear una GPO que va a permitir la conexión desde el servidor a los equipos y que permitirá recibir la orden de actualizar las GPOs entre otras cosas. Para tener ordenados los equipos, vamos a crear una "OU" llamada "equipos" y dentro de ella otras "OU" para cada departamento que queramos. Los equipos que tengamos en la red los deberemos ordenar en estas Unidades Organizativas creadas, y de esta manera podremos mandar la actualización a los equipos que nos interese.

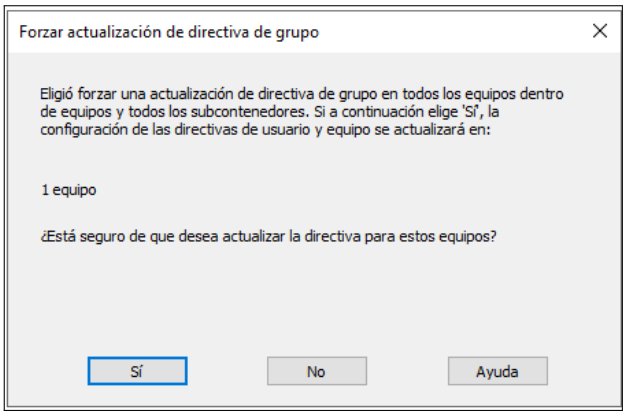
Vamos a crear una nueva GPO llamada "Permitir actualizar GPOs desde servidor" y la vamos a crear en la "OU" de "equipos".

La ruta para crear es: "*Configuración del equipo → Directivas → Plantillas Administrativas → Red → Conexiones de red → Firewall De Windows Defender → Perfil de dominio →* ***Firewall de Windows Defender: permitir excepción de administración remota entrante***", la cual habilitaremos pero sólo permitiendo la IP de nuestro servidor. El resumen de la GPO sería el que aparece en la imagen:

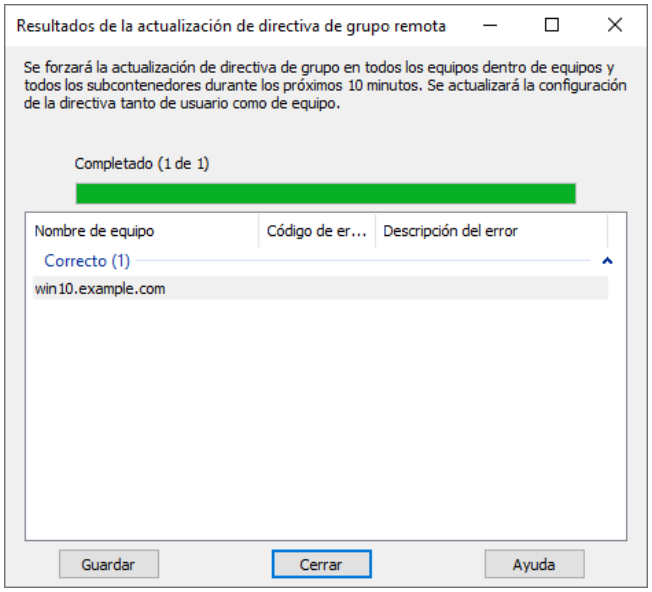


Para aplicar esta GPO el equipo remoto deberá reiniciarse. A partir de este momento, cuando ya se haya aplicado la GPO podremos forzar la actualización de las GPO desde el servidor desde el propio "Administrador de directivas de grupo", haciendo click derecho sobre la OU donde están los equipos que nos interesa y dando a "Actualización de directiva de grupo"





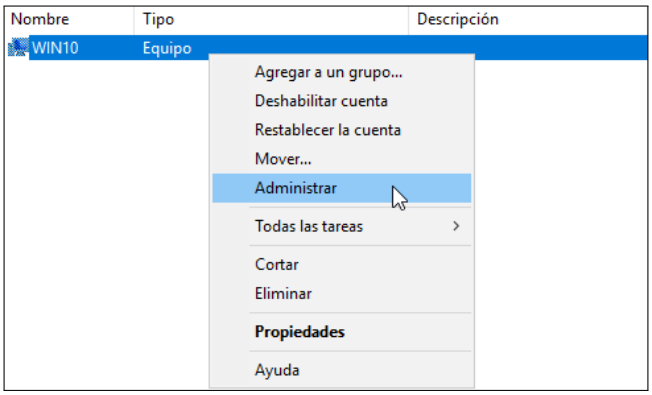
Tras confirmar que queremos actualizar las directivas para el número de equipos que estén en esa Unidad Organizativa, nos aparecerá un mensaje de confirmación de que el proceso se ha realizado de manera correcta:



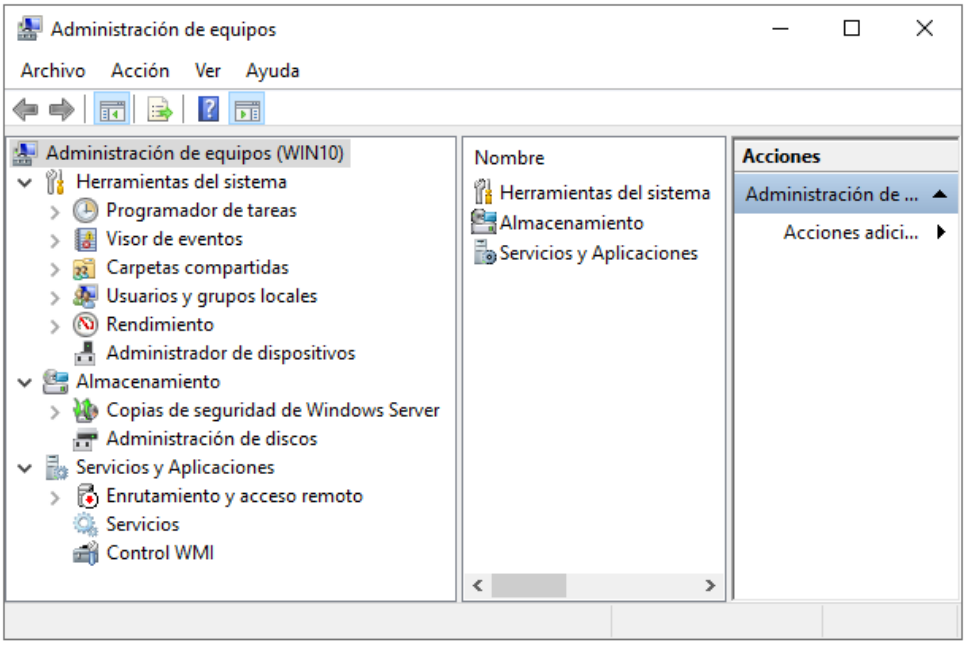
Debemos fijarnos que en la columna central no aparece ningún código de error. Este proceso hará que las GPOs se recarguen en una tarea que se va a ejecutar en el equipo remoto en los **próximos 10 minutos**. Si

queremos que sea inmediato podemos forzar la actualización mediante un comando en powershell.

También podremos administrar los equipos desde el programa de "Usuarios y equipos de Active Directory", eligiendo el equipo que queramos y darle a "Administrar":



Lo que nos abrirá el asistente de administración remota:



Y desde aquí podremos realizar parte de la administración remota del equipo.

6.2.4. Habilitar acceso al escritorio remoto

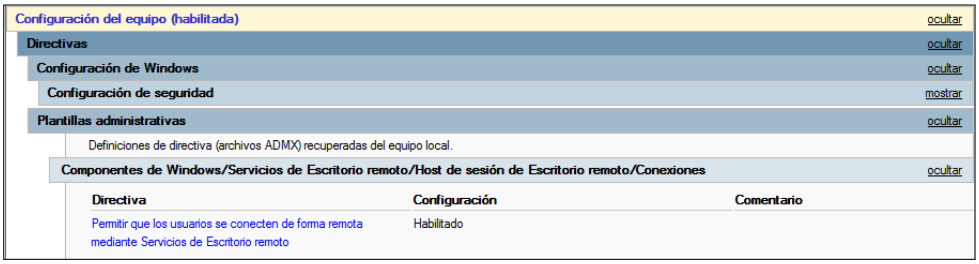
En una red no siempre tenemos acceso a los equipos de los usuarios, por encontrarse en otras sedes, oficinas o situaciones geográficas. Es por esto por lo que el poder acceder a los equipos de los usuarios **supone una ventaja administrativa a nivel global**.

Para ello, podemos crear una GPO para que los equipos que están en nuestro Active Directory se les active el acceso remoto para poder así conectarnos a ellos.

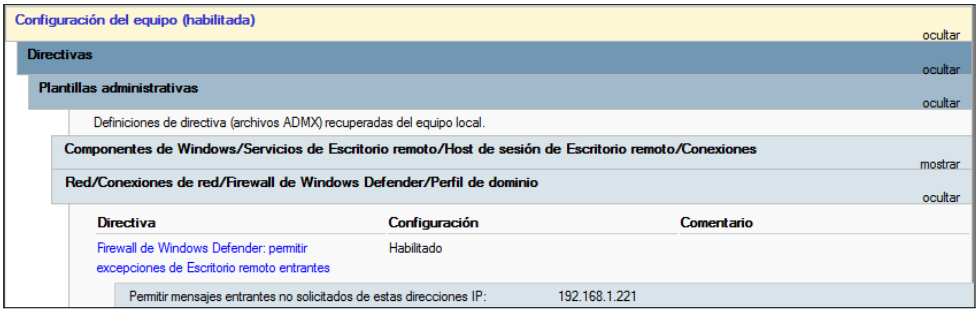
Los pasos a dar cuando creamos la GPO son:

1. **Habilitar escritorio remoto**: "*Configuración del equipo → Directivas → Plantillas administrativas → Componentes de Windows → Servicios de Escritorio remoto → Host de sesión de Escritorio remoto →*

Conexiones → Permitir que los usuarios se conecten de forma remota mediante Servicios de Escritorio remoto"



2. Habilitar el acceso en el firewall al puerto necesario (RDP, que es 3389). Existe una plantilla para ello en: "*Configuración del equipo → Directivas → Plantillas administrativas → Red → Conexiones de Red → Firewall de Windows Defender → Perfil de dominio*" y aceptar conexiones desde la IP del servidor.



Información

Por defecto sólo los usuarios administradores pueden realizar conexiones de acceso remoto.

3. Si queremos que cualquier otro usuario que no sea el Administrador pueda realizar conexiones mediante escritorio remoto, crearemos un grupo nuevo, por ejemplo "**escritorio remoto**".

   Este grupo debe ser admitido en los equipos remotos dentro del grupo local "**Usuarios de escritorio remoto**". Esto lo haremos a través de la GPO utilizando la directiva: "*Configuración del equipo → Preferencias → Configuración del Panel de control → Usuarios y grupos locales*". Hacemos click derecho, para crear nuevo grupo local, y seleccionaremos:

   - **Nombre de grupo**: Usuarios de escritorio remoto (integrado).
   - **Miembros**: El nuevo grupo que hemos creado

¡Atención!

Recuerda que para que funcione tienes que forzar que se actualice la GPO de manera remota o con "gpupdate.exe".

7. Alta Disponibilidad en Active Directory

Cuando hablamos de Alta Disponibilidad hacemos referencia al diseño de una infraestructura de servicios y de servidores en la que no existe un único punto de fallo. Hasta ahora, nuestro dominio está controlado por un único servidor que realiza la función de Controlador de Dominio, por lo que en caso de catástrofe, perderíamos la posibilidad de configurar el dominio, añadir nuevos usuarios…

Para evitar ese único punto de fallo, es conveniente realizar la instalación de un nuevo servidor y configurarlo para que actúe como servidor réplica del controlador de dominio principal.

7.1. Instalación del servidor réplica

La instalación del servidor réplica será igual que la del primer controlador de dominio, tal como hemos visto anteriormente. Lógicamente nos aseguraremos que el servidor cuente con otro nombre y otra dirección IP que no coincida con el servidor principal.

También realizaremos la instalación del rol de "**Servicios de dominio de Active Directory**" ya que lo vamos a necesitar más adelante.

7.2. Meter servidor en el dominio actual

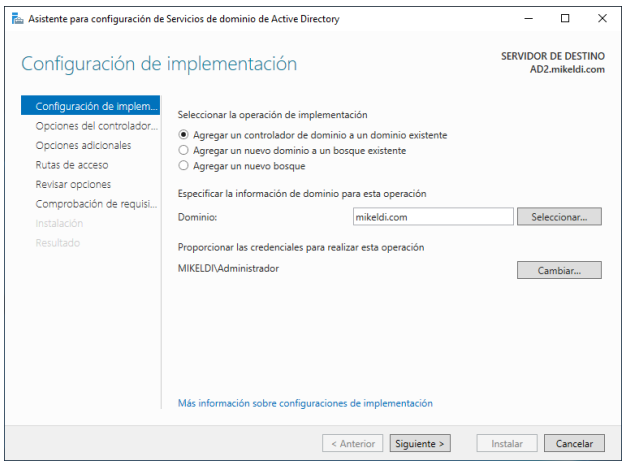
El primer paso es meter el nuevo servidor como un equipo más en el dominio del que queremos ser réplica. Los pasos a dar son los vistos anteriormente para un equipo Windows 10.

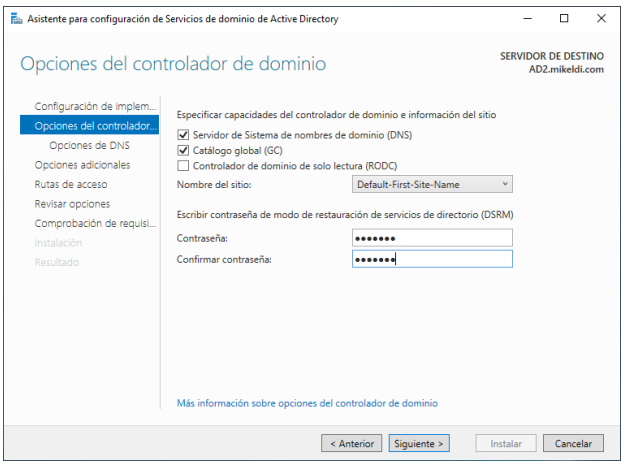
Tras realizar el reinicio correspondiente, vamos a promocionar el nuevo servidor para que actúe como servidor réplica.

7.3. Configurar nuevo Active Directory como servidor réplica

Para promocionar el nuevo servidor Active Directory para que actúe como servidor réplica del dominio debemos ir al "Administrador del servidor" y pinchar sobre el aviso para promocionarlo, tal como hicimos en el servidor principal.

Al darle a promocionar, nos aparecerá en la nueva ventana la opción de "**Agregar un controlador de dominio a un dominio existente**" seleccionada, con el dominio correspondiente al que nos hemos metido previamente. Le daremos a "Cambiar" y meteremos las credenciales de Administrador del dominio. De esta manera, tendremos la ventana tal como aparece en la siguiente imagen y podremos dar a "Siguiente".

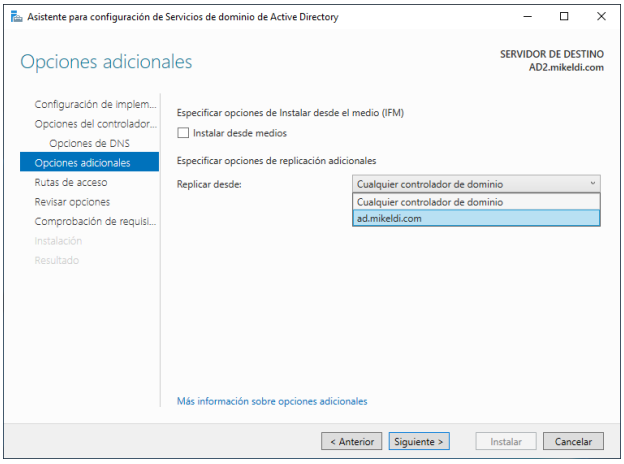




En este paso podremos elegir si queremos que nuestro servidor sea configurado en modo “sólo lectura” (por lo que realizará la réplica de datos, pero no podremos crear nada en él) y nos preguntará por la contraseña DSRM para este servidor. Una vez introducida, continuamos hasta llegar al siguiente paso.

En este paso nos permitirá realizar las opciones del DNS que necesitemos. En nuestro caso, vamos a continuar al siguiente paso sin realizar modificaciones en este apartado, por lo que llegaremos a las “opciones adicionales”:

Tal como se puede ver en la captura de pantalla, podremos elegir desde qué servidor queremos realizar la réplica de los datos.

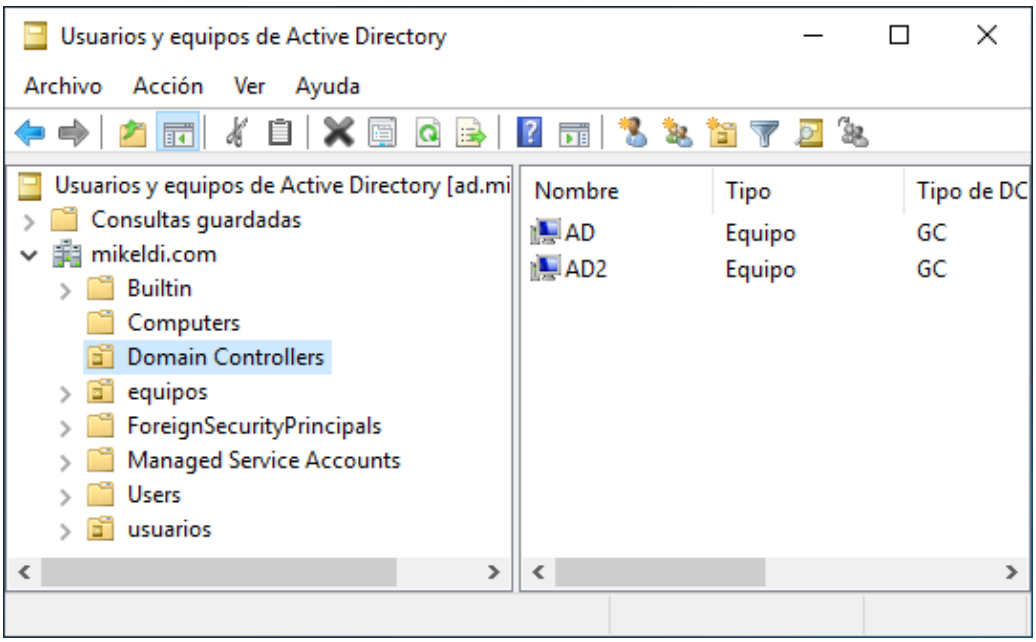


En nuestra infraestructura sólo tenemos otro servidor, por lo que no importa si seleccionamos el servidor o la opción “Cualquier controlador de dominio”. Esta opción es útil cuando ya tengamos previamente dos o más servidores, ya que por conectividad nos puede interesar que realice la réplica desde uno en concreto.

Los siguientes pasos los dejaremos tal como están hasta realizar la instalación y que el servidor se reinicie.

7.4. Comprobación del funcionamiento de la réplica

Una vez reiniciado el servidor réplica, veremos que al iniciar sesión nos aparecerán los grupos, usuarios y GPOs que teníamos en el servidor principal. Si vamos al gestor de “Usuarios y equipos de Active Directory” veremos que el nuevo servidor aparece dentro del apartado “Domain Controller”:

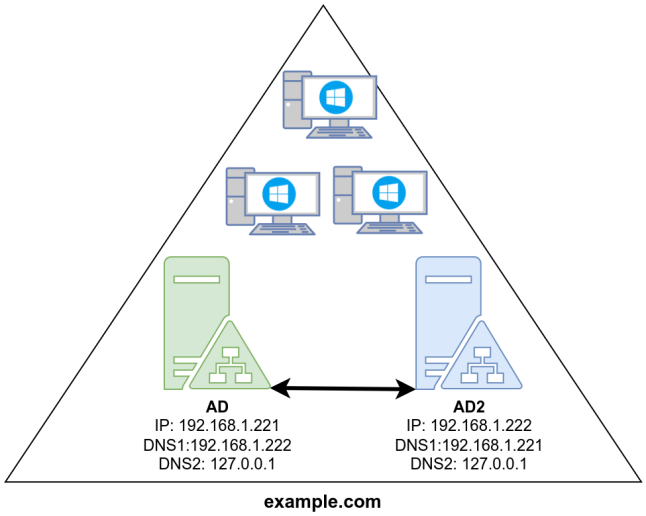


Si creamos un nuevo usuario en cualquiera de los dos servidores, comprobaremos cómo en el otro servidor nos aparece también. De ser así, la réplica está funcionando de manera correcta.

7.5. Configuración final

Para dejar nuestra configuración final correcta debemos de realizar una pequeña modificación en los DNS de los servidores y también en los clientes.

El cambio de configuración es que los servidores tienen que tener al otro servidor como DNS principal, y a sí mismos como DNS secundarios. Es decir, configuración cruzada:



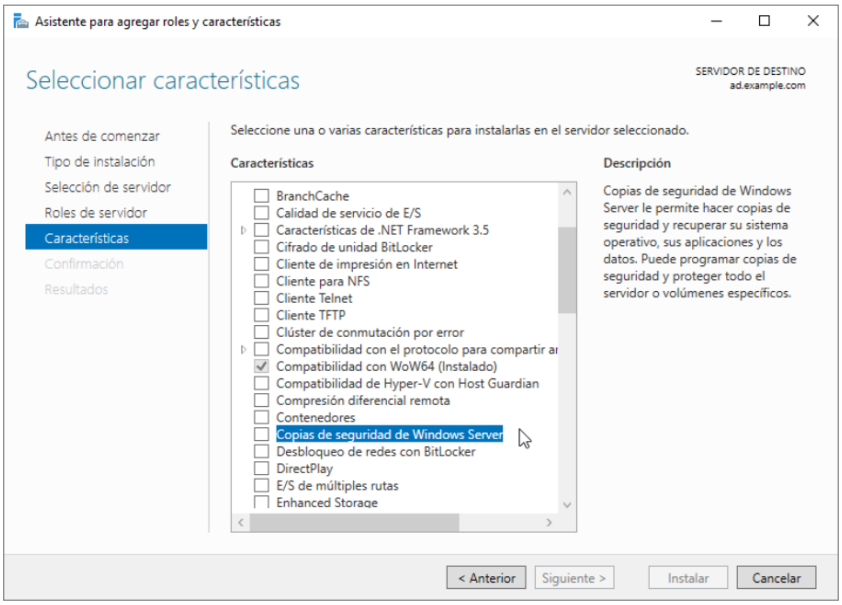
Tal como se puede ver en la imagen, contamos con dos servidores para el dominio “example.com”, ambos se replican entre sí y cada uno tiene como DNS principal al otro servidor.

Por otro lado, en lo referente a los equipos, la configuración debe ser igual: deben de tener como DNS principal uno de los servidores y como DNS secundario el otro . De esta manera, si el servidor que actúa como DNS principal cae, los equipos realizarán las peticiones al otro servidor. Esta modificación de DNS lo

habitual es que se realice mediante el servidor DHCP en lugar de ir equipo por equipo (ya que es una tarea tediosa y fácilmente automatizable).

8. Copias de seguridad en Windows Server 2022

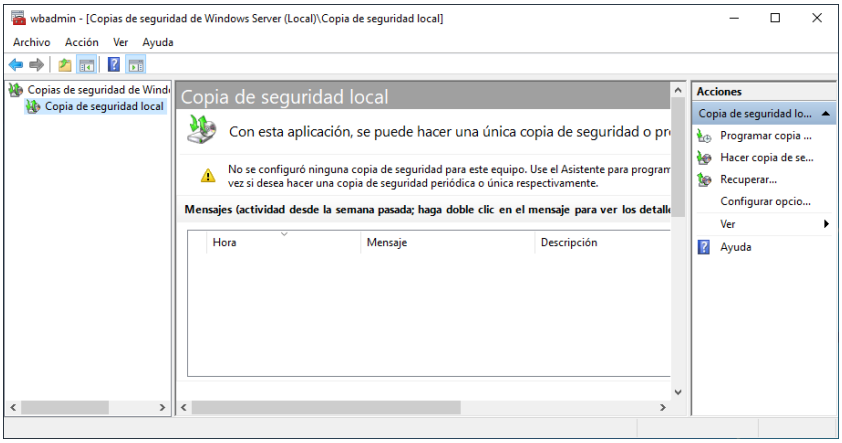
Windows Server 2022 no tiene instalado por defecto la característica de copias de seguridad, por lo que deberemos instalarla. Para ello abriremos el "Administrador del Servidor", le daremos a "Agregar Roles y Características", iremos hasta el apartado de características y seleccionaremos la opción de "Copias de seguridad de Windows Server":



Una vez instalado podremos comenzar con el proceso de copias de seguridad. Pero antes, deberemos haber realizado una serie de pasos.

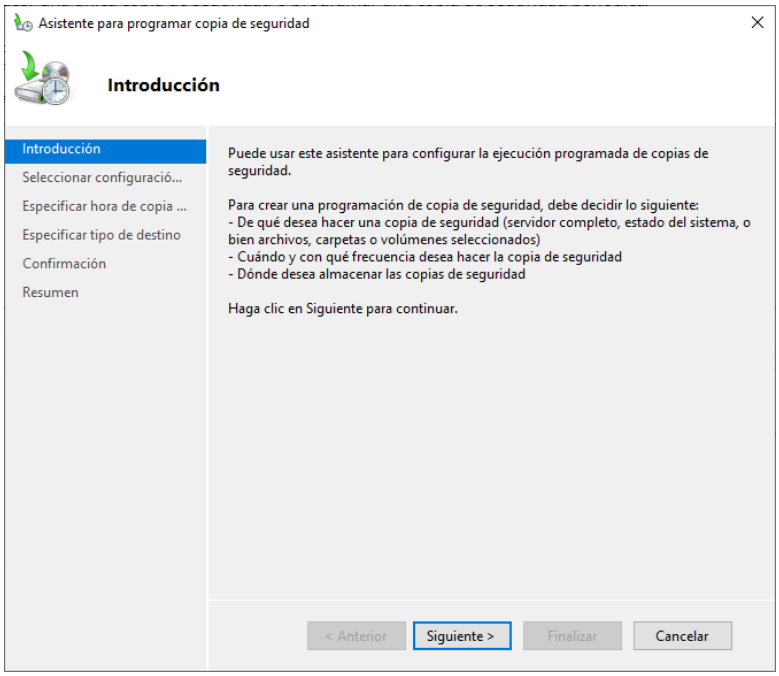
8.1. Configurar copia de seguridad en Windows Server

Una vez hecha la instalación, podemos ejecutar el asistente de copias de seguridad que está situado en la carpeta de "Herramientas Administrativas".



Ahora ya podremos configurar el tipo de copia de seguridad que necesitemos. Para ello haremos click en la acción de "Programar copia de seguridad" o haciendo click derecho encima de "Copia de seguridad local",

y nos abrirá un asistente con una serie de pasos:



Este asistente nos permitirá hacer dos tipos de copias de seguridad:

- **Servidor completo**: Esta opción nos permitirá realizar una copia de todo el servidor, de las aplicaciones que están instaladas y del estado del sistema
- **Personalizada**: En caso de elegir esta opción, podremos elegir volúmenes, directorios o archivos de los que queremos guardar una copia de seguridad.

Una vez elegido el tipo de copia, podremos **elegir la hora** a la que queremos que se haga el backup, y si queremos que sea una vez al día o varias veces.

Posteriormente podremos elegir el volumen de destino donde guardaremos la copia de seguridad. El asistente de Windows Server nos deja tres opciones:

- **Disco duro dedicado**: Este es el método más fiable cuando realizamos copias de seguridad locales. Para ello necesitaremos tener un segundo disco duro en el servidor, que sólo será accesible por el programa de copias de seguridad.
- **En un volumen**: Esta opción no es recomendada, ya que de perderse el volumen, perdemos los datos y la copia de seguridad.
- **En una carpeta de red compartida**: es el método ideal para realizar copias de seguridad que no sea en local. El problema que trae consigo es que sólo serán accesibles los datos de la última copia de seguridad.

Una vez terminado el asistente el programa configurará la tarea de realizar las copias de seguridad a las horas elegidas. También podremos forzar copias de seguridad “de una vez” en cualquier momento.

8.2. Restauración de copia de seguridad

Como hemos dicho previamente, tras una copia de seguridad podremos realizar la restauración de los datos cuando así sea necesario. También hemos comentado previamente que es recomendable que cada cierto tiempo deberíamos realizar la restauración de datos en un servidor de pruebas para confirmar que las copias de seguridad se están realizando de manera correcta.

En el caso de las copias de seguridad realizadas en Windows Server podremos realizar dos tipos de restauraciones:

- **Parcial**: Nos permitirá realizar una restauración de determinadas opciones a elegir, y podremos realizar la restauración en el mismo servidor, u en otra ruta que deseemos. Podremos restaurar:
  - **Archivos y carpetas**: podremos recorrer la estructura de ficheros y restaurar los ficheros o carpetas que nos interesen.
  - **Volúmen**: podríamos recuperar un volumen completo como puede ser "C:".
  - **Aplicaciones**: restaurar aplicaciones que estuviesen instaladas
  - **Estado del sistema**: el estado del sistema completo tal como estaba en el momento de la copia de seguridad.
- **Completa**: Este es el caso más catastrófico, y sería en caso de rotura del disco duro original de la instalación. En este caso, tendríamos que coger el disco duro del backup, meterlo en otro servidor, introducir la ISO de Windows Server y en lugar de darle a "Instalar Ahora" darle a "Reparar el equipo".

9. Comandos PowerShell para Windows Server

En Windows 7 y Windows Server 2008 R2 Microsoft añadió una auténtica interfaz de consola (en inglés command-line interface o **CLI**) que permite la ejecución de comandos para poder administrar tanto un equipo de usuario como un servidor. Esta interfaz es conocida como PowerShell y viene instalado en todos los equipos Windows desde entonces.

Esta interfaz de consola está diseñada para su uso por parte de administradores de sistemas, con el propósito de automatizar tareas o realizar scripts con tareas repetitivas. Como curiosidad, PowerShell es Software Libre y se puede descargar y utilizar también en Mac OS y Linux.

En este apartado vamos a hacer uso de distintos comandos que nos van a ser útiles para tener información acerca del estado del Servidor, del Bosque, del Dominio, de usuarios de nuestro Active Directory y de las GPO que hayamos creado.

Para obtener ayuda y saber cómo se ejecuta, así como los parámetros que tienen los comandos, podremos hacer uso del comando **get-help**. Por ejemplo, para obtener la ayuda del comando Get-ADForest deberíamos hacer:

Obtener ayuda del comando Get-ADForest

```
get-help Get-ADForest
```

Tras ejecutar este comando es posible que nos pregunte si queremos actualizar los sistemas de ayuda (se descarga los manuales de los comandos).

9.1. Sobre Active Directory

A continuación vamos a ver comandos que tienen que ver con el Active Directory propiamente dicho:

A continuación vamos a ver comandos que tienen que ver con el Active Directory propiamente dicho.

Listar todos los comandos posibles a ejecutar que tienen que ver con Active Directory:

Listar todos los comandos sobre Active Directory

```
get-command -Module ActiveDirectory
```

Información del Bosque:

Información del bosque

```
Get-ADForest
```

Información del Dominio que tiene el controlador de dominio:

Información del bosque

```
Get-ADDomain
```

9.2. Sobre usuarios

Comandos que tienen que ver con los usuarios creados dentro del Active Directory.

Listar todos los usuarios del Active Directory:

Listar usuario del Active Directory

```
Get-ADUser -filter *
```

Listar todos los usuarios del Active Directory, pero sólo obtener los nombres:

Listar sólo los nombres de usuarios del Active Directory

```
Get-ADUser -filter * | select name
```

Listar todos los usuarios del dominio **example.com** del Active Directory:

Listar sólo los usuarios de un dominio Active Directory

```
Get-ADUser -filter * -SearchBase "dc=example,dc=com"
```

Listar todos los usuarios de un OU concreto (tiene que existir esa unidad organizativa) dentro del dominio example.com (en este caso) Active Directory:

Listar sólo los usuarios de una OU del Active Directory

```
Get-ADUser -filter * -SearchBase "OU=usuarios,dc=example,dc=com"
```

Crear el usuario “usuario1” (como no tiene contraseña, se creará deshabilitado):

Crear usuario

```
New-ADUser usuario1
```

Crear el usuario “usuario2” con la contraseña “MIk3ld1”:

Crear usuario con la contraseña

```
New-ADUser usuario2 -AccountPassword(ConvertTo-SecureString "P4s5w0rd"
-AsPlainText -Force) -Enabled $true
```

Crear el usuario “usuario3” con la contraseña “MIk3ld1” y que al hacer login le pida cambiar la contraseña:

Crear usuario con la contraseña y opciones extra

```
New-ADUser usuario2 -AccountPassword(ConvertTo-SecureString "P4s5w0rd"
-AsPlainText -Force) -Enabled $true -ChangePasswordAtLogon $true
```

Listar todos los usuarios cuyo usuario contengan “usu”:

Listar usuarios

```
Get-ADUser -filter {name -like "*usu*"}
```

Listar información interna de un usuario:

Obtener información de un usuario

```
Get-ADUser usuario1
```

Listar los usuarios deshabilitados:

Obtener usuarios deshabilitados

```
Search-ADAccount -AccountDisabled | select name
```

Deshabilitar al usuario “usuario3”:

Deshabilitar un usuario

```
Disable-ADAccount -Identity usuario3
```

Habilitar al usuario “usuario3”:

Habilitar un usuario

```
Enable-ADAccount -Identity usuario3
```

Listar usuarios bloqueados:

Listar usuarios bloqueados

```
Search-ADAccount -LockedOut
```

Forzar el cambio de contraseña al usuario “usuario1” en el siguiente login:

Forzar cambio de contraseña

```
Set-ADUser -Identity usuario1 -ChangePasswordAtLogon $true
```

9.3. Sobre grupos

Comandos que tienen que ver con los grupos creados dentro del Active Directory.

Listar todos los grupos de un Active Directory:

Listar grupos

```
Get-ADGroup -Filter *
```

Listar información de un grupo:

Listar información de un grupo

```
Get-ADGroup usuarios_oficina
```

Listar usuarios de un grupo:

Listar usuarios de un grupo

```
Get-ADGroupMember -Identity usuarios_oficina | select name
```

Añadir usuarios de un grupo (existe la opción de importar usuarios de un fichero CSV):

Añadir usuarios a un grupo

```
Add-ADGroupMember -Identity usuarios_oficina -Members user1, user2
```

9.4. Sobre ordenadores

Comandos que tienen que ver con los ordenadores asociados al Active Directory.

Listar los ordenadores que pertenecen a un Active Directory:

Listar ordenadores de un Active Directory

```
Get-AdComputer -filter *
```

Listar ordenadores con Sistema Operativo Windows 10 de un Active Directory:

Listar ordenadores con Windows 10 en un AD

```
Get-AdComputer -filter  {OperatingSystem -Like '*Windows 10*'}
```

Apagar ordenador, incluso aunque haya un usuario conectado:

Apagar un ordenador de manera remota

```
Stop-Computer -ComputerName WIN10 -Force
```

Reiniciar ordenador, incluso aunque haya un usuario conectado:

Reiniciar un ordenador de manera remota

```
Restart-Computer -ComputerName WIN10 -Force
```

9.5. Sobre GPO

Listar todos los comandos posibles a ejecutar que tienen que ver con las GPO (group policy objects):

Posibles comandos sobre GPOs

```
get-command -Module grouppolicy
```

Listar todos los GPO (group policy objects):

Posibles comandos sobre GPOs

```
get-gpo -All
```

Crear un fichero HTML en c: con las configuraciones de todas las GPO:

Todas las configuraciones de GPOs en formato HTML

```
get-GPOreport -Domain example.com -ALL ReportType Html -path c:\gpo.html
```

Forzar la actualización inmediata de las GPOs en el equipo remoto “win10”

Forzar actualización de GPOs de manera remota

```
Invoke-GPUpdate -Computer win10 -Force -RandomDelayInMinutes 0
```

9.6. Otros comandos

Información del sistema:

Información del sistema

```
systeminfo
```

Listar todos los procesos que están corriendo:

Listar todos los procesos

```
Get-Process
```

Listar los servicios:

Listar todos los servicios

```
Get-Service
```

Comprobar conexión al puerto 3389 del equipo remoto win10:

Comprobar puerto remoto

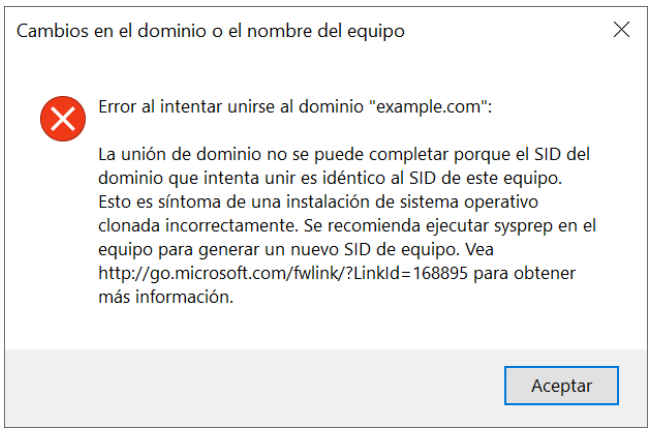
```
Test-NetConnection win10 -CommonTCPPort 3389
```

10. Sysprep - Preparando nuestro sistema

Durante la instalación de Windows (tanto la versión Server como de escritorio), se genera un identificador de seguridad (**SID**, de *Security IDentifier*), y otra información específica del PC que debe ser única dentro de nuestra infraestructura.

Como esa información debe ser única por equipo, **afecta a la hora de querer crear una imagen de sistema que nos sirva como base para realizar una instalación masiva en toda nuestra red**.

Por ejemplo, si intentamos meter un equipo clonado a un Active Directory nos dará el siguiente error:

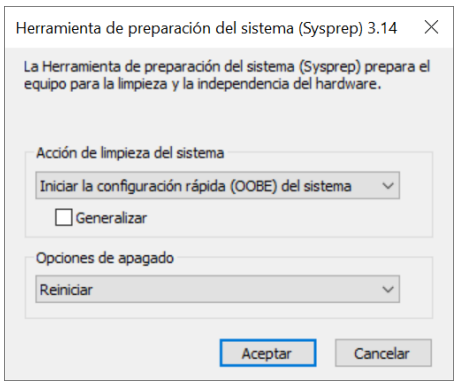


Es por eso que Microsoft incorpora la aplicación **Sysprep**, que **se encarga** de eliminar esa información específica. Se encuentra en la ruta "`%WINDIR%\system32\sysprep\sysprep.exe`" y puede ser ejecutado tanto en consola como en entorno gráfico.

Las funciones principales que nos permite Sysprep son:

- Borra la información específica del PC de la imagen de Windows, incluyendo el SID. Esto permite crear imágenes que pueden ser aplicadas a otros equipos.
- Desinstala (pero sin borrar) drivers específicos del equipo.
- Prepara el PC para ser arrancado en modo **OOBE** (*Out-of-Box Experience*). Este sistema deja el equipo con la instalación preparada como si un usuario arrancase por primera vez.
- Permite configuración "desatendida", añadiendo un fichero con las respuestas de configuración.

La manera más sencilla de ejecutarlo es yendo a la ruta mediante el explorador de Windows y haciendo doble click:



Tal como se puede ver, **Sysprep** nos ofrece una serie de opciones que podemos elegir en su ejecución:

- **Acción de limpieza del sistema**:
  - **Iniciar la configuración OOBE**. Comentada previamente.
  - **Iniciar modo auditoria del sistema**. El sistema auditoría permite saltarse el OOBE para entrar con permisos de Administrador para realizar configuraciones e instalaciones en la imagen. Más información
  - **Generalizar**: Esta es la opción que elimina la información específica del equipo.

Información

Si queremos crear una imagen o usar una máquina clonada tenemos que usar la opción generalizar.

- **Opciones de apagado**: Qué hacer una vez ejecutado Sysprep.

II

GNU/Linux

1. Introducción a GNU/Linux

1.1. Un poco de historia

Para conocer cómo nació el movimiento GNU y el kernel Linux debemos conocer un poco de historia de la informática y cómo evolucionó en los primeros años.

1.1.1. El nacimiento de Unix

1964-1969 Los laboratorios **Bell** empiezan un proyecto con el **MIT** (Instituto Tecnológico de Massachusetts) y **General Electric** para desarrollar un sistema de **tiempo compartido** ("time-sharing computing"): se llamaría **Multics** (Multiplexed Information and Computing Service).

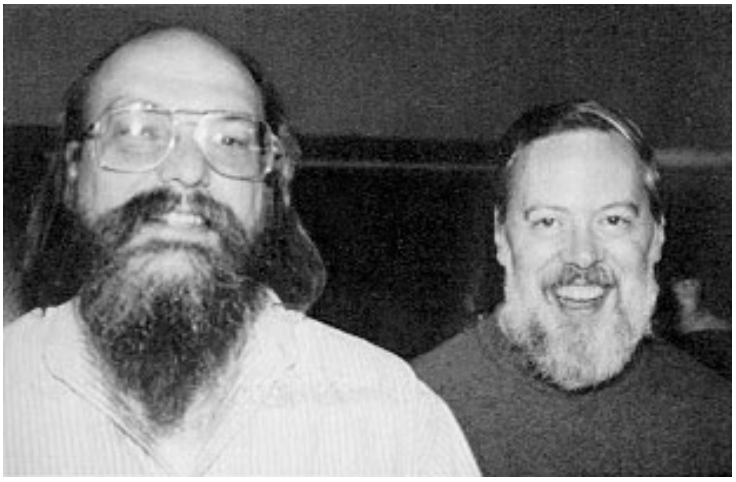
Hasta este momento, los sistemas utilizados eran de un único proceso, la CPU no era compartida por múltiples procesos sino que se ejecutaba por lotes (se les mandaba los procesos a ejecutar y se ejecutaban en orden).

Multics obtuvo licencia libre en el 2007. En Diciembre del 2016 salió la última versión 12.6f.

1969 Uno de los desarrolladores de Multics, Ken Thompson, decidió escribir su propio sistema operativo. Ken Thompson es conocido también por crear el lenguaje de programación **B**, el sistema de codificación de caracteres UTF-8 y el lenguaje de programación Go, entre otras cosas.

A Ken Thompson se le une Dennis Ritchie y otros, y empiezan a programar un sistema de ficheros jerárquico, el concepto de procesos de computación, ficheros de dispositivos, un intérprete de comandos, … El resultado de lo programado era más pequeño y simple que Multics, lo que se convertiría en Unix. En Agosto ya tendrían el sistema operativo, se auto-gestiona, tenía un assembler, un editor y una shell de comandos.

Dennis Ritchie es conocido también por crear junto con Ken el lenguaje de programación **C** (aparece por primera vez en 1972).



Ken y Dennis. Origen: Wikipedia

1970 En ese momento el nuevo sistema operativo se llamaba **Unics** (*Uniplexed Information and Computing*

Service, un juego de palabras en contraposición a Multics). No tenían todavía dinero de la organización en el desarrollo (era desarrollado por los programadores) y tampoco era multitarea todavía.

A finales de año el sistema ya era conocido como **UNIX**, y se había portado a la máquina PDP-11.

Las primeras versiones de Unix incluían el código fuente para que las universidades lo pudiesen modificar y así poder extenderlo a sus necesidades.

1971 El sistema se empieza a hacer complejo y como querían que más usuarios lo usasen, crean el sistema de manuales que es utilizado hoy en día (mediante el comando **"man"**).

Dennis Ritchie y Ken Thompson trabajando en un PDP-11. Origen: Wikipedia.

1973 La versión 4 del sistema es reescrita completamente en C. Hasta este momento el sistema había estado escrito en ensamblador, por lo que no era portable entre distintos tipos de máquinas, aunque la primera versión portada a otra plataforma fue en 1978. Se cree que había "más de 20" instalaciones del sistema.

1974 La versión 5 se licencia para ser utilizada en **instituciones educativas**.

1975 La versión 6 se licencia para poder ser utilizadas por empresas por $20.000 de la época.

1977 La universidad de Berkeley lanza su primera versión de Unix bajo la Berkeley Software Distribution (BSD).

1979 Con la salida de Unix v7, se comienza a portar a los distintos "microordenadores" de la época y a los distintos microprocesadores (Motorola 68000, Intel 8086, …).

1980 Microsoft anuncia su primer Unix para microcomputadoras de 16 bits (Xenix).

1.1.2. El nacimiento de GNU (GNU's Not Unix)

1971 Richard Stallman comienza su carrera en el MIT en el laboratorio de inteligencia artificial.

Es conocido no sólo por el movimiento GNU, si no también por crear GCC y Emacs entre otra gran cantidad de software.

En esa época el software se distribuía de manera abierta para poder ser modificado. Lo habitual era realizar modificaciones para mejorar el software y distribuirlo entre compañeros y universidades.

1982 Richard Stallman quiere modificar el firmware de unas impresoras y el fabricante le pide que firme un acuerdo de no divulgación si le enseñan el código. Esto hace que Stallman se enfurezca y es cuando decide que la situación actual debe cambiar y volver al sistema de intercambio de software anterior.

Richard Stallman: Wikimedia

1983 Se anuncia el nacimiento del proyecto **GNU**, cuya finalidad es la de construir un sistema operativo completamente libre, compatible con Unix. La idea es dar a los usuarios la libertad y el control de sus ordenadores.

1985 Se lanza el manifiesto GNU, y ya cuenta con un editor de texto (Emacs), compilador de C, una shell, varias utilidades … El núcleo inicial todavía no es funcional.

1986 Richard Stallman escribe y publica la definición de lo que es Free Software (Software Libre) a través de la Free Software Foundation.

Más adelante veremos a qué se refiere sobre libertad en el software.

1.1.3. El nacimiento de Minix

1987 Andrew S. Tanenbaum crea Minix como propósito educativo y para enseñar cómo funciona un sistema operativo.

1991 Sale la versión 1.5 de Minix y es portada a distintas arquitecturas (IBM, Motorola 68000, Amiga, Apple Macintosh, …).

1992 Debate con Linus Torvalds sobre la arquitectura del kernel Linux (núcleo monolítico) en lugar de usar un micronúcleo.

1.1.4. El nacimiento de Linux

1991 Un estudiante en la universidad de Helsinki, Linus Torvalds, comienza un proyecto personal escrito para su nuevo ordenador, un PC con procesador 80386.

El desarrollo comienza bajo **Minix**, usando el compilador **GCC** del movimiento GNU (GCC = GNU Compiler Collection).

El proyecto termina convirtiéndose en un kernel de un sistema operativo y escribió al grupo de noticias de Minix diciendo:

"Hola a todos los que estáis ahí fuera usando minix.

Estoy haciendo un sistema operativo (libre), (solamente por aficion, no será grande ni profesional como el GNU) para clones 386(486) AT.

…

PD. Sí – está libre de cualquier código de minix, y tiene un sistema de ficheros multi-hilo. NO es portable (usa el cambio de tareas del 386 etc), y probablemente nunca soporte otra cosa que no sean los discos duros AT, porque es todo lo que tengo :-(. "

Linus torvalds. Origen: Wikipedia

1992 Originalmente la licencia de Linux era propia e impedía el uso comercial de Linux. En la versión 0.99 esto cambia y se cambia a la licencia GNU Public License (**GPL**).

1993 El proyecto cuenta con más de 100 desarrolladores. El kernel se adapta al entorno del proyecto GNU. Nace la distribución **Debian** (una de las más importantes a día de hoy)

Debian

1994 Se libera la versión 1.0. El proyecto XFree86 se une y Linux consigue interfaz gráfico. Nacen las primeras distribuciones comerciales **Red Hat** y **Suse**.

1998 Empresas como **IBM**, **Compaq** y **Oracle** anuncian que apoyan a Linux. Nace el interfaz gráfico **KDE**.

1999 Nace el interfaz gráfico **GNOME** como reemplazo a KDE, ya que KDE hacía uso de una librería propietaria en aquel momento (QT).

2001 Steve Ballmer (CEO de Microsoft) dice: **"Linux es un cáncer"**.

2002 Se libera OpenOffice (originalmente suite ofimática de Sun Microsystems). Nace Mozilla (hoy día: Firefox).

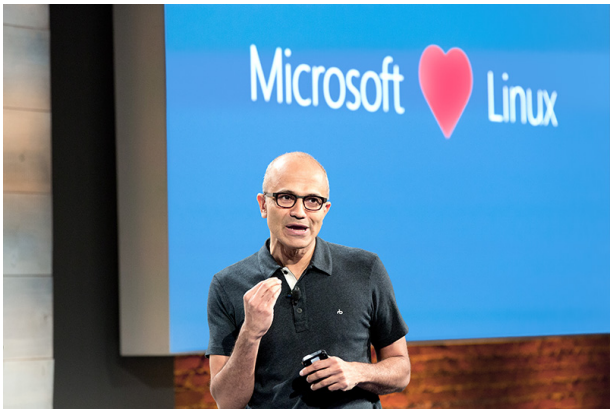
2003 IBM lanza un anuncio para la Linux Foundation: https://www.youtube.com/watch?v=x7ozaFbqg00

2004 Nace **Ubuntu** (basándose en Debian) y Steve Ballmer (CEO de Microsoft) dice que Linux infringe muchas de sus patentes.

2008 Nace **Android**, sistema operativo con kernel Linux. Actualmente es el sistema operativo de móviles que más terminales tiene.

2009 Red Hat iguala a Sun Microsystem en capitalización bursátil (un gran logro simbólico).

2014 Satya Nadella (CEO de Microsoft) muestra en una presentación la siguiente transparencia:

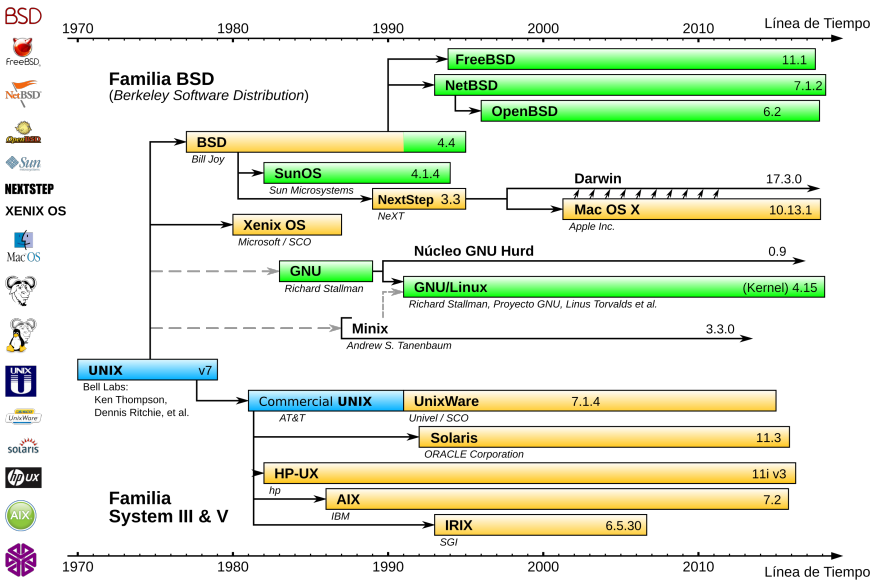


Origen: Wikipedia

2016 Microsoft anuncia WSL (*Windows Subsystem for Linux*) y se puede instalar en Windows 10 y Windows Server 2019. Permite correr ejecutables de Linux nativamente.

1.1.5. Cronograma de sistemas Unix

En el siguiente cronograma se puede ver la línea temporal de los sistemas Unix:



Origen: Wikipedia

1.2. Resumen

Linux es conocido como un sistema operativo libre pero el nombre de Linux se centra única y exclusivamente en el **kernel** (o **núcleo**) del sistema operativo.

El sistema operativo completo debería llamarse **GNU/Linux**, ya que el kernel es una "pequeña" parte (aunque muy importante) dentro de todo el sistema operativo. El resto de herramientas utilizadas en el sistema operativo pertenecen al proyecto GNU.

2. Licencias Libres

2.1. Free Software / Software Libre

En 1986 Richard Stallman saca a la luz la definición de lo que es Free Software (Software Libre) a través de la Free Software Foundation:

Información

La palabra "free" no se refiere a gratis, se refiere a libertad.

(*The word free in our name does not refer to price; it refers to freedom.*)

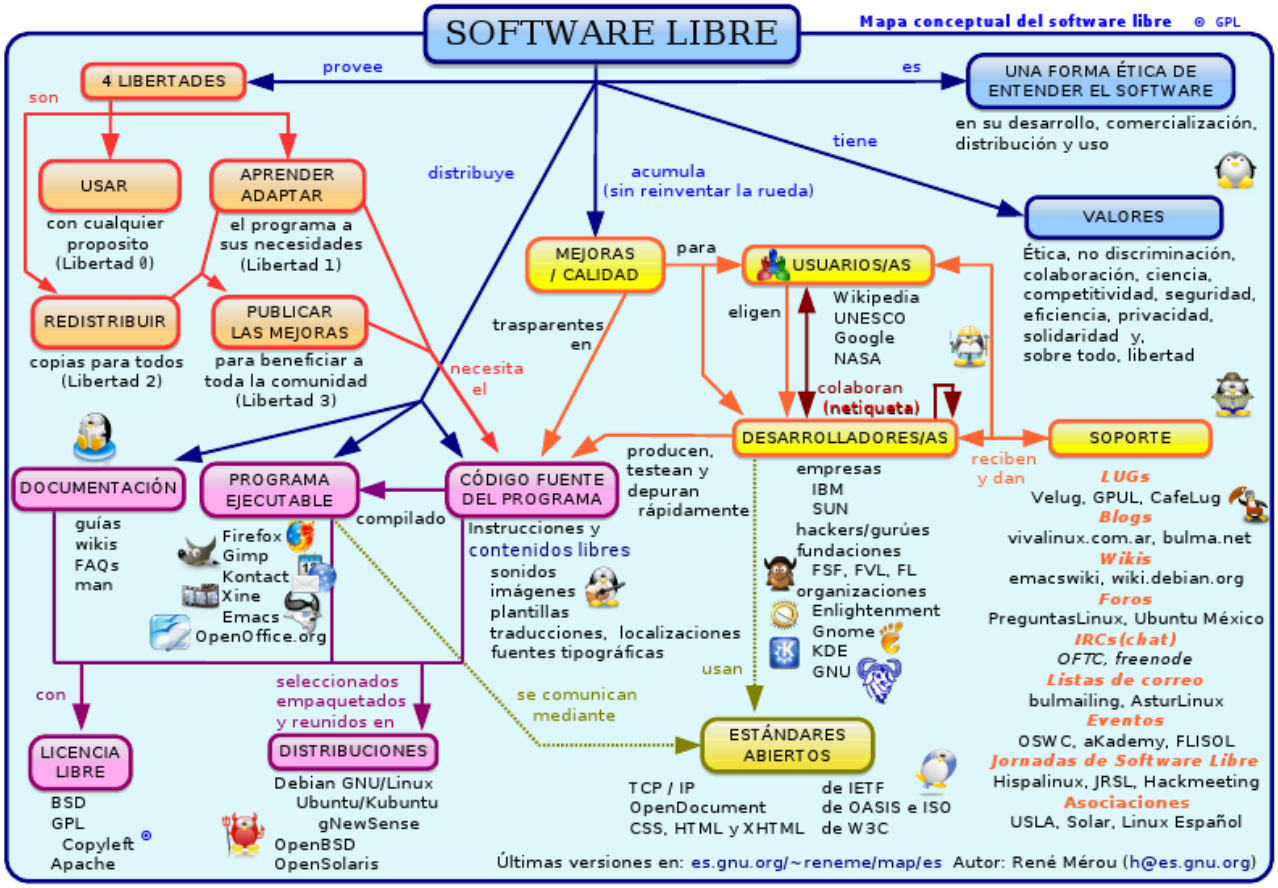
Las libertad en el software se refiere a:

1. La libertad de ejecutar el programa, **para cualquier propósito**.
2. La libertad de **estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera**. El acceso al código fuente es una condición necesaria para ello.
3. La libertad de **redistribuir copias** para que pueda ayudar al prójimo.
4. La libertad de **mejorar el programa y publicar sus mejoras, y versiones modificadas en general**, para que se beneficie toda la comunidad. El acceso al código fuente es una condición necesaria.

El movimiento del Free Software es un movimiento que tiene que ver más con la filosofía y la ética que con la tecnología en sí misma.

2.1.1. Copyleft y GNU Public License (GPL)

Es una práctica legal que consiste en el ejercicio del derecho de autor (copyright en inglés) con el objetivo de propiciar el libre uso y distribución de una obra, exigiendo que los concesionarios preserven las mismas libertades al distribuir sus copias y derivados (Wikipedia).



Mapa conceptual del Software Libre: Wikipedia

Con esto nació la licencia GNU GPL, la cual permite al usuario final la libertad de usar, estudiar, compartir y modificar el software recibido. Tiene que quedar claro que un programa comercial puede ser Software Libre.

2.1.2. Diferencias con el Open Source

Los programas Open Source son aquellos que podemos ver el código fuente pero esto no quiere decir que podamos modificarlo o adaptarlo a nuestras necesidades.

El Open Source es menos restrictivo que el Software Libre y se puede decir que todo Software Libre es Open Source, pero no todo Open Source tiene por qué ser libre.

2.1.3. Licencias libres más conocidas

Un listado de las licencias libres más utilizadas (en la Wikipedia existe una tabla comparativa):

- GNU GPL
- BSD
- MIT
- Licencia Apache
- Licencia PHP
- Creative Commons (no todas las versiones). Más utilizadas en contenido multimedia.

3. Sistema de ficheros en GNU/Linux

El sistema de ficheros en GNU/Linux, al igual que en Unix, es jerárquico, comenzando en la raíz denominada "/". Partiendo de esta raíz, el resto del sistema de ficheros nace en forma de ramificaciones generando lo que se denominan "rutas de ficheros", que es el camino completo para llegar al mismo.

3.1. Filesystem Hierarchy Standard #{fhs}

Debido a que en GNU/Linux todo se representa como ficheros (discos, dispositivos, programas, …) es necesario que exista un orden a la hora de ser almacenados. Con esa intención nace en 1993 el estándar de la jerarquía de ficheros de Linux, enfocado a reestructurar los archivos. Posteriormente se unieron otros derivados de UNIX (la comunidad de desarrollo de BSD) por lo que terminó adoptando el nombre FHS.

Aún siendo un estándar, no todas las distribuciones lo siguen al pie de la letra, y otros Unix, como MacOS, tienen sus propias rutas especiales.

3.2. Directorios importantes

A continuación se exponen los directorios más importantes del sistema junto con la descripción del contenido que deben de tener:

- **/boot/**: archivos de arranque del kernel, normalmente junto con la configuración utilizada para compilarlos.
- **/dev/**: contiene archivos especiales de bloque que representan los dispositivos del hardware que está corriendo el sistema operativo
- **/etc/**: contiene los archivos de configuración del servidor y de los servicios que corren en él. Está subdividido en directorios por servicios o configuraciones.
- **/home/**: los directorios de trabajo de los usuarios normales del sistema
- **/lib/**: librerías que hacen funcionar a los programas
- **/root/**: es la home del usuario root
- **/var/**: archivos variables del sistema
    - **/var/lib/**: aquí se suelen guardar los ficheros de los programas que "crecen": bases de datos, ficheros caché…
    - **/var/log/**: los logs del sistema

Junto a todos estos directorios, se ha separado los lugares en los que van los binarios, o ejecutables de los programas. Lo habitual es que se encuentren en estas rutas:

- **/bin/**: aplicaciones esenciales del sistema
- **/sbin/**: aplicaciones que en principio sólo debería ejecutar el usuario root o programas de administración del sistema

- **/usr/bin/**: ejecutables de usuario
- **/usr/sbin/**: ejecutables de superusuario

Aunque las rutas de los ejecutables denotan quién debería ejecutar el programa, en la vida real no tiene por qué ser una limitación.

3.3. Dispositivos de almacenamiento y discos duros

En sistemas operativos Windows es habitual que cada partición cuente con una letra para acceder a ella, al igual que ocurre cuando introducimos un dispositivo de almacenamiento externo (un pendrive).

Tal como se ha comentado, en sistemas Unix el sistema de ficheros es una jerarquía, y por tanto todo dispositivo de almacenamiento nuevo deberá estar montado bajo la raíz "/". Hoy día, en distribuciones con escritorio, al introducir un pendrive éste es auto-montado (es accesible) desde la ruta **/media/**, donde aparecerán tantos directorios como discos hayamos conectado.

3.3.1. Almacenamiento permanente

Si queremos que un disco duro nuevo sea permanente en nuestro sistema, podremos montarlo en cualquier lugar de la estructura jerárquica. Debido a este sistema, el usuario final no se tendrá que preocupar en almacenar los ficheros en una ruta distinta, si no que será el administrador el que haya hecho que esa ruta ahora pertenezca a un disco duro nuevo.

Imaginemos que el sistema operativo se ha instalado en un disco duro pequeño de 32Gb de espacio y se está llenando, y el directorio que más ocupa es el directorio de los usuarios. Podremos añadir al servidor un nuevo disco duro montado en /home y por tanto a partir de ahora los datos guardados en /home estarán en un nuevo disco duro más grande.

Ejemplo de discos en un sistema con "lsblk"

```
root@vega:~# lsblk
NAME                      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda                         8:0    0   1,8T  0 disk
└─sda1                      8:1    0   1,8T  0 part /home/backup

sdb                         8:16   0   3,6T  0 disk
└─sdb1                      8:17   0   3,6T  0 part /home/disco4tb
sdc                         8:32   0 447,1G  0 disk
├─sdc1                      8:33   0   529M  0 part
├─sdc2                      8:34   0   100M  0 part
├─sdc3                      8:35   0    16M  0 part
└─sdc4                      8:36   0 446,5G  0 part
nvme0n1                   259:0    0 931,5G  0 disk
```

```
├─nvme0n1p1                 259:1    0   512M  0 part
└─nvme0n1p2                 259:2    0   800G  0 part /home
nvme1n1                     259:3    0 931,5G  0 disk
├─nvme1n1p1                 259:4    0   512M  0 part /boot/efi
├─nvme1n1p2                 259:5    0    90G  0 part /
├─nvme1n1p3                 259:6    0   300G  0 part
│ ├─VMs-ubuntu--20.04--so1 254:0    0    10G  0 lvm
│ ├─VMs-manjaro            254:2    0    20G  0 lvm
│ └─VMs-win10              254:3    0    35G  0 lvm
└─nvme1n1p4                 259:7    0 156,2G  0 part
```

4. Gestión de usuarios locales en GNU/Linux

En las distribuciones GNU/Linux lo habitual suele ser que existan al menos dos usuarios tras una instalación:

- **root**: usuario administrador o súper usuario.
- **usuario no-privilegiado**: durante la instalación de la distribución nos suele preguntar para crear un usuario del sistema, que no tendrá privilegios.

El usuario root, como se ha dicho previamente, es el administrador del sistema, tiene permisos para realizar cualquier tarea dentro de nuestro sistema: instalar paquetes, desinstalarlos, modificar cualquier fichero, realizar formateos... Por lo tanto, el **realizar tareas como usuario root puede ser peligroso si cometemos algún fallo**.

Las buenas prácticas nos dicen que las tareas cotidianas del sistema deberíamos realizarlas como usuario normal y **sólo convertirnos en root cuando sea estrictamente necesario**.

4.1. Creación de usuarios locales

Tras instalar el sistema, veremos que se nos han creado varios usuarios en el sistema, aparte del usuario **root** y el usuario **no-privilegiado**. Para poder ver los usuarios que existen en nuestro sistema podemos verlo en el fichero `/etc/passwd` o podríamos obtener un listado ejecutando el siguiente comando:

Listar usuarios del sistema

```
root@vega:~# cut -d: -f1 /etc/passwd
```

Para crear un usuario:

Crear usuarios del sistema

```
root@vega:~# adduser ruben

Añadiendo el usuario `ruben' ...
```

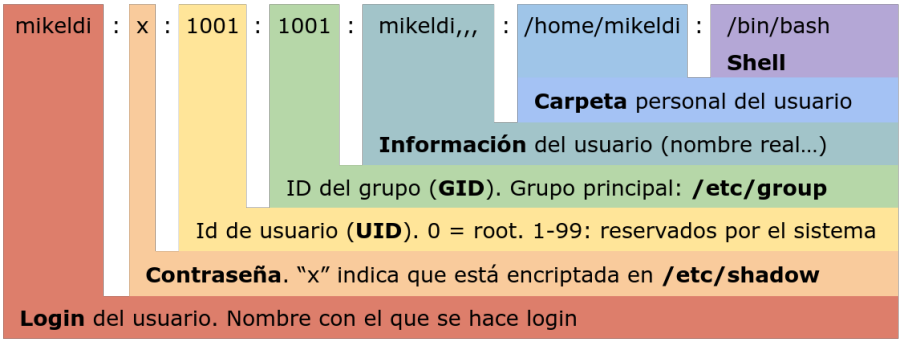
```
Añadiendo el nuevo grupo `ruben' (1001) ...
Añadiendo el nuevo usuario `ruben' (1001) con grupo `ruben' ...
Creando el directorio personal `/home/ruben' ...
Copiando los ficheros desde `/etc/skel' ...
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para ruben
Introduzca el nuevo valor, o pulse INTRO para usar el valor predeterminado
    Nombre completo []:
    Número de habitación []:
    Teléfono del trabajo []:
    Teléfono de casa []:
    Otro []:
¿Es correcta la información? [S/n]
```

Y la línea que nos creará en el fichero `/etc/passwd` es:

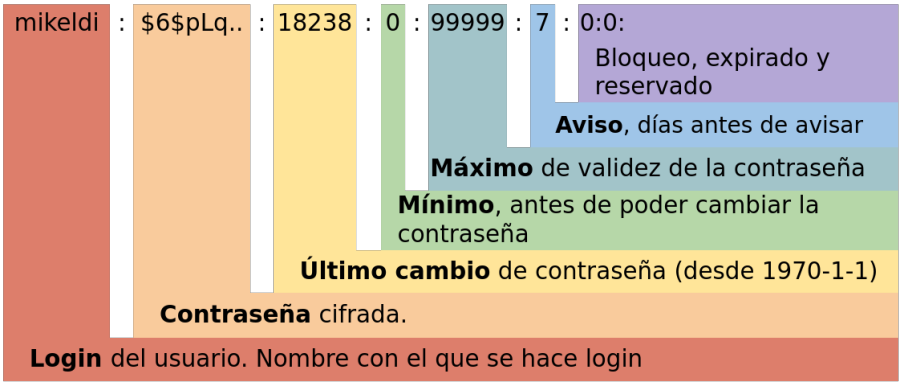
Línea de un usuario en /etc/passwd

```
ruben:x:1001:1001:ruben,,,:/home/ruben:/bin/bash
```

El fichero `/etc/passwd` nos muestra los datos de los usuarios, siendo un fichero que tiene distintos datos separados por ":", siendo cada apartado:



En las primeras versiones GNU/Linux la contraseña de los usuarios aparecía en el propio fichero /etc/passwd, lo que suponía un problema en la seguridad, ya que no estaban cifradas. Actualmente, las contraseñas de los usuarios se almacenan cifradas en el fichero `/etc/shadow`. El fichero es similar al passwd, estando separados los apartados por ":"



En el apartado de la contraseña podemos saber cierta información acerca de la misma ya que tiene el siguiente formato: **"idsalt$hashed"**

- **id**: el algoritmo utilizado para cifrar la contraseña
  - 1 – MD5
  - $2a$ – Blowfish
  - $2y$ – Eksblowfish
  - 5 – SHA-256
  - 6 – SHA-512

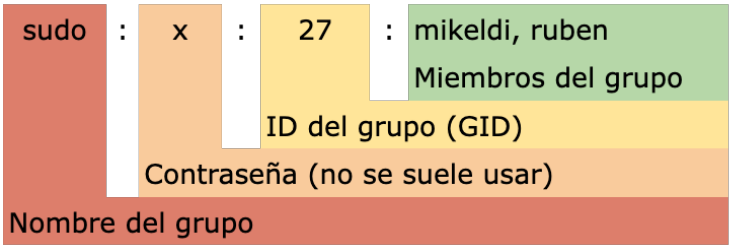
Aparte, también podemos encontrarnos con:

- **Contraseña vacía**: Si no hay contraseña, al pedirnos la contraseña a la hora de hacer login será suficiente con pulsar "intro".
- **!**, *****: la cuenta está bloqueada para la contraseña. El usuario no podrá loguearse utilizando la contraseña. Resulta útil si queremos bloquear el acceso con contraseña pero no con otros métodos (clave pública SSH).
- ***LK***: cuenta bloqueda. El usuario no podrá loguearse.
- ***NP***, **!!**: Nunca se ha puesto una contraseña

4.2. Gestión de grupos

En algunas distribuciones GNU/Linux, al crear un usuario directamente nos crea un grupo para el nuevo usuario. En otras, el usuario pertenece al grupo "users".

Para saber los grupos a los que pertenece un usuario podemos ejecutar el comando `groups`. Los grupos del sistema aparecen en el fichero `/etc/group`, y al igual que los ficheros vistos previamente, están separados por "**:**".



4.3. Permisos de ficheros

En GNU/Linux los ficheros cuentan con 3 tipos de permisos:

- lectura (**r**ead): el usuario puede leer el fichero
- escritura (**w**rite): el usuario puede escribir en el fichero
- ejecución (e**x**ecute): el usuario puede el fichero o puede ver el contenido de un directorio

Todos ello para los distintos usuarios que pueden existir en el sistema:

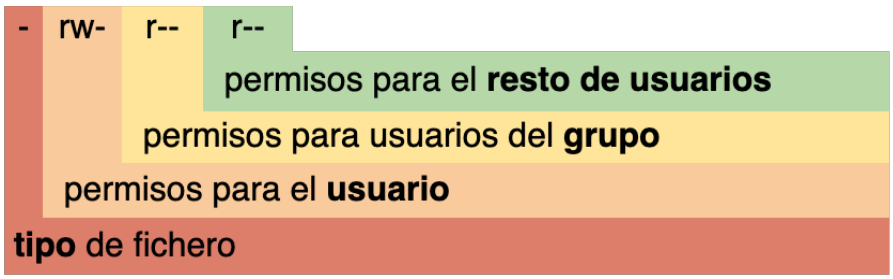
- **dueño del fichero**: la persona que ha creado el fichero
- **grupo**: los usuarios pertenecientes al grupo al que pertenece el fichero tendrán ciertos privilegios
- **el resto de usuarios**: los permisos que tendrán el resto de usuarios que no son ni el dueño ni pertenecen al grupo

Todo ello se puede visualizar en el sistema de ficheros si listamos los permisos del fichero:

Ver los permisos de un fichero

```
ruben@vega:~$ ls -lh fichero.txt
-rw-r--r-- 1 ruben ruben 0 dic  8 19:17 fichero.txt
```

Los permisos se pueden ver en los primeros 10 caracteres:



Existen los distintos tipos de ficheros:

- **-**: fichero normal
- **d**: directorio
- **b**: dispositivo de bloque (ejemplo: /dev/sda*)
- **c**: dispositivo de carácter (las consolas. ejemplo: /dev/tty*)

- **s**: socket local
- **p**: tubería (pipe)
- **l**: enlace simbólico (link)

4.3.1. Permisos especiales

Existen otros permisos especiales:

- **SUID**: permiso especial que permite que el fichero sea ejecutado con los permisos del dueño del fichero (aunque lo ejecute otro usuario). Se visualiza con una "S" en el permiso de ejecución del dueño `-rwsrw-r--`.
- **SGID**: permiso especial que permite que el fichero sea ejecutado como el grupo. Aparece una "S" en el permiso de ejecución del grupo: `-rwxr-s---`.
- **STICKY**: si el bit sticky está activado en un **directorio** sólo el usuario root, el dueño del directorio o el dueño del fichero puede borrar ficheros dentro de dicho directorio. Aparece una "t" en el permiso de ejecución del resto de usuarios: `drwxrwxrwt`. Muchas distribuciones usan este permiso en el directorio `/tmp`

4.3.2. Cambiando permisos y dueños a los ficheros y a los directorios

Para cambiar los permisos a los ficheros y a los directorios se hace con el comando **chmod**.

Para cambiar permisos de dueño a los ficheros y a los directorios se hace con el comando **chown**.

4.4. La importancia de "sudo"

En muchas distribuciones GNU/Linux el usuario no-privilegiado que se crea tiene permiso de "sudo" para poder ejecutar comandos como si se tratara del **root** (u otro usuario) para poder realizar tareas de administración. Es habitual que en estas distribuciones **el usuario root no suela tener contraseña**.

Cuando un usuario necesite realizar una tarea como administrador, deberá usar "sudo" antes del comando:

Editar un fichero con permisos de root

```
ruben@vega:~$ sudo nano /etc/passwd
```

Tras realizar este comando, el sistema nos pedirá la contraseña del usuario con el que lo estemos ejecutando y comprobará que el usuario tiene permisos de "sudo" para poder ejecutar el comando (en este caso: nano).

El comando "**sudo**" viene de "**su**per user **do**" (que en inglés sería: "super usuario haz"), y aunque su uso habitual es el de permitir realizar cualquier comando de administración, la configuración permite mucho más, pudiendo permitir a ciertos usuarios sólo realizar ciertas tareas. Por ejemplo:

- Usuario **ruben**: tendría permisos para poder realizar cualquier comando del sistema.
- Usuario **dba**: sólo tendría permisos para poder realizar el reinicio del sistema de base de datos.

- Usuario **adminweb**: sólo tendría permisos para poder realizar el reinicio del servidor web.
- …

De esta manera, la gestión de nuestro servidor estaría basada en múltiples usuarios y cada usuario sólo sería capaz de realizar pequeñas tareas, por lo que la seguridad del servidor sería mayor y limitaría lo que los usuarios puedan realizar.

4.4.1. Configurando "sudoers"

Los permisos de sudo se realizan en el fichero `/etc/sudoers`, y para su edición se hace uso del comando **visudo**, el cual abre el fichero y se asegura que a la hora de guardar la sintaxis es correcta.

Si realizamos cualquier modificación sobre el fichero, éste será tenido en cuenta la próxima vez que se realice la ejecución del comando "sudo", por lo tanto, no hay que realizar ningún reinicio de servicio.

El fichero `/etc/sudoers` tiene permisos de sólo lectura para el usuario root y el grupo root:

Permisos del fichero /etc/sudoers

```
root@vega:# ls -lh /etc/sudoers
-r--r----- 1 root root 669 jun  5  2017 /etc/sudoers
```

Un fichero sudoers suele tener el siguiente aspecto:

Contenido del fichero /etc/sudoers

```
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

La línea que más importa en este fichero es la que indica "**%sudo ALL=(ALL:ALL) ALL**" y es explicada en su comentario anterior. Lo que quiere decir es que cualquier usuario que pertenezca al grupo "sudo" podrá realizar cualquier comando del sistema como superusuario. La sintaxis de la línea es:

- **%sudo**: cualquier usuario que pertenezca al grupo "sudo"
- **ALL**= : desde cualquier host o IP

- **(ALL:ALL)**: el usuario que ejecuta puede ejecutar el comando como cualquier usuario y cualquier grupo
- **ALL**: puede ejecutar cualquier comando

Un ejemplo limitando el uso de sudo a un único comando a un usuario:

Añadimos configuración al fichero /etc/sudoers

```
ruben    ALL=(ALL:ALL) NOPASSWD:/bin/systemctl suspend
```

Con esta línea lo que estamos permitiendo es que el usuario "ruben" puede ejecutar el comando "/bin/systemctl suspend" (suspender el equipo) y sin necesidad de meter contraseña al hacer sudo, gracias a la opción "NOPASSWD").

4.5. Diferencias entre "sudo", "su" y "su -"

Como ya se ha comentado en el apartado anterior, "sudo" permite la ejecución de comandos como cualquier usuario, siendo lo habitual ejecutarlo como root. Ahora bien, en entornos donde el usuario root tiene contraseña, nos puede interesar convertirnos en él para realizar tareas sin tener que estar ejecutando "sudo" a cada comando. Al ser root, tendremos que tener especial cuidado.

4.5.1. Variables de entorno

En cualquier sistema operativo existen las denominadas "variables de entorno". Son variables que cada usuario tiene y sirven para indicar ciertos parámetros que se están utilizando (la SHELL que se está usando), o parámetros que se van a usar a la hora de ejecutar comandos o realizar tareas, ya que se consultan a ellas. En GNU/Linux las variables de entorno se pueden consultar ejecutando:

Vemos las variables de entorno del usuario ruben

```
ruben@vega:~$ printenv
LANG=es_ES.utf8
LOGNAME=ruben
XDG_VTNR=2
COLORTERM=truecolor
PWD=/home/ruben
DESKTOP_SESSION=gnome
USERNAME=ruben
SHELL=/usr/bin/zsh
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
...
```

Una variable de entorno puede consultarse haciendo:

Consultamos el contenido de la variable PATH

```
ruben@vega:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

Como se puede ver, es con un "**$**" y el nombre de la variable en mayúsculas. Existen muchas variables de entorno, y podríamos crear las nuestras propias si así lo necesitáramos.

4.5.2. La importancia de "su -"

Con el comando "**su**" nos podemos convertir en cualquier otro usuario del sistema siempre y cuando **conozcamos su contraseña**. Hay que notar la diferencia respecto a "**sudo**" que cuando lo ejecutamos nos pide **nuestra contraseña**.

Al ejecutar "su" nos convertimos en el usuario root (o ejecutando "su usuario2", nos convertimos en el usuario2), **pero no hacemos uso de sus variables de entorno**, si no que seguimos con las variables de entorno del usuario que éramos previamente. Para convertirnos en el usuario y que obtengamos sus variables de entorno es necesario ejecutar "**su -**", o lo que es lo mismo: "**su -l**", que el manual nos dice: "*Start the shell as a login shell with an environment similar to a real login*". Por ejemplo:

Consultamos el contenido de la variable PATH en distintas situaciones

```
ruben@vega:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games

ruben@vega:~$ su
Contraseña:

root@vega:/home/ruben# echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games

root@vega:/home/ruben# exit

ruben@vega:~$ su -
Contraseña:

root@vega:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

El usuario "ruben" tiene unos valores en la variable de entorno PATH (es la variable que se encarga de tener las rutas de los ejecutables de los programas). Al convertirse en root haciendo uso de "su", y mirar la variable PATH, podemos observar que es igual que el usuario prueba.

Ahora bien, si a la hora de convertirse en root hace uso de “su -”, se puede ver cómo la variable PATH obtiene otros valores, siendo lo más significativo que aparecen las rutas “/usr/local/sbin” y “/usr/sbin” que son las rutas donde se almacenan los ejecutables que (en principio) sólo deberían ejecutarse como administrador del sistema.

5. SAMBA

5.1. Introducción

Samba es una implementación libre del protocolo de archivos compartidos que es utilizado por los sistemas Microsoft Windows para sistemas de tipo UNIX. De esta forma, es posible que ordenadores con GNU/Linux, Mac OS X o Unix en general se vean como servidores o actúen como clientes en redes de Windows.

Samba también permite validar usuarios haciendo de Controlador Principal de Dominio (PDC), como miembro de dominio e incluso como un dominio Active Directory para redes basadas en Windows.

Samba fue desarrollado originalmente para Unix por Andrew Tridgell utilizando un sniffer, o capturador de tráfico, para entender el protocolo usando **ingeniería inversa**.

5.1.1. SMB/CIFS

Server Message Block (**SMB**) y Common Internet File System (**CIFS**) son protocolos de red desarrollados para compartir archivos e impresoras entre nodos de una red. El protocolo SMB fue desarrollado originalmente por IBM y posteriormente ampliado por Microsoft y renombrado como CIFS.

Los términos SMB y CIFS son a menudo intercambiables pero hay características en la implementación de SMB de Microsoft que no son parte del protocolo SMB original. Sin embargo, desde una perspectiva funcional, ambos son protocolos utilizados por Samba.

La versión 1 del protocolo SMB no debería usarse, y en Samba está deshabilitado por defecto, debido a los ataques que hubo por el ransomware Wannacry. Estos ataques utilizaban una vulnerabilidad en servidores que no estaban parcheados.

5.2. Instalación

Para instalar Samba haremos uso de los repositorios oficiales de nuestra distribución, y por tanto, podremos instalarlo haciendo uso del siguiente comando:

Instalamos SAMBA y parte de las dependencias

```
root@vega:~# apt install samba cifs-utils smbclient winbind
```

Tras la instalación, nos habrá creado un directorio de configuración `/etc/samba`, cuyo fichero de configuración principal es `/etc/samba/smb.conf`. Tras la instalación, tendremos un fichero de configuración estándar que no podrá realizar demasiado, ya que la configuración es escasa.

5.3. Comprobar configuración

Como ya se ha comentado, la configuración principal está en `/etc/samba/smb.conf`, con una configuración estándar, por lo que tendremos que adecuarla a nuestras necesidades.

Para asegurar que las modificaciones que hemos realizado son correctas, podremos hacer uso del siguiente comando:

Comprobar configuración de Samba

```
root@vega:~# testparm
```

Es aconsejable ejecutarlo antes de realizar ningún reinicio del servicio, ya que de haber algún error el servicio se quedaría parado, y por tanto dejaríamos sin servicio a los usuarios.

Una vez realizado los cambios y comprobado que el fichero no contiene ningún error (al menos a nivel de sintaxis), reiniciamos el servicio de la siguiente manera:

Reiniciar servicio de Samba

```
root@vega:~# systemctl restart smbd
```

El problema con el comando anterior es que Samba cuenta con varios servicios: smbd, nmbd, winbindd… por lo que para evitar que se nos olvide reiniciar uno, o asegurar que toda la configuración se recarga, lo mejor es hacer:

Recargar toda la configuración de Samba

```
root@vega:~# smbcontrol all reload-config
```

Una vez reiniciada la configuración, podremos listar los servicios que está compartiendo y/o usando Samba con el siguiente comando:

Resumen de la configuración de Samba

```
root@vega:~# smbclient -L localhost
```

5.4. Samba como servidor “standalone”

Samba tiene distintos modos de funcionar, siendo el sistema “standalone” el método por defecto que suele estar configurado en las distribuciones GNU/Linux.

Cuando el servidor está en este modo puede correr por sí solo o puede unirse a dominios de Windows. Es el método más sencillo de funcionar y **es utilizado para compartir carpetas de manera sencilla**.

Este modo “standalone” suele ser utilizado cuando tenemos pocos usuarios y sólo queremos compartir carpetas, como puede ser en casa o una oficina. En el momento en el que tenemos varios usuarios y queremos que nuestro servidor se convierta en un Active Directory **no tendremos que utilizar este modo**.

5.5. Samba como Controlador de Dominio

Desde la versión 4, se permite configurar Samba como un servidor de dominio al más puro estilo Active Directory siendo compatible con él, por lo que podremos hacer que nuestros equipos con Windows Pro se puedan conectar al dominio creado en Samba.

5.5.1. Preparando el servidor

Antes de realizar la configuración del Dominio tenemos que realizar una serie de modificaciones para dejar el servidor preparado.

5.5.1.1. Poner IP estática

Tal como hicimos con Windows Server, todo servidor debe de tener puesta una IP estática para poder realizar sus funciones de manera correcta. Para realizar la configuración de IP estática en Ubuntu los pasos están en el anexo.

5.5.1.2. Modificación del DNS

Ubuntu cuenta con un servidor DNS propio configurado en el sistema que nos puede causar problemas a la hora de ejecutar Samba y que entre en servicio su propio DNS, por lo tanto lo que se va a realizar en este apartado es desactivar el DNS de Ubuntu. En otras distribuciones (como sucede en Debian) este proceso no es necesario al no contar con un servidor DNS instalado previamente.

Los pasos para parar y desactivar el DNS son:

Paramos y deshabilitamos el servicio DNS “resolved”

```
root@vega:~# systemctl stop systemd-resolved
root@vega:~# systemctl disable systemd-resolved
```

De no ejecutar estos comandos, Samba nos dará errores durante la ejecución. Estos comandos hacen que el DNS actual se pare y se deshabilite para que no se vuelva a activar en el siguiente arranque.

¡Cuidado!

¡Reiniciamos el servidor! Así será más fácil realizar los siguientes pasos.

Tras reiniciar, tenemos que hacer que nuestro servidor utilice el servicio DNS que va a levantar Samba. Para ello, tenemos que borrar el fichero `/etc/resolv.conf` (que realmente es un enlace simbólico creado por el anterior servicio), volverlo a crear y asegurarnos que aparece lo siguiente:

Configuración del fichero /etc/resolv.conf

```
nameserver 127.0.0.1
```

5.5.1.3. Configurar fichero /etc/hosts

Antes de realizar la configuración del Dominio, tenemos que realizar una nueva configuración en el fichero `/etc/hosts`, para que se vea reflejado lo siguiente:

Configuración del fichero /etc/hosts

```
127.0.0.1       localhost
192.168.1.200   dc.midominio.com dc
```

Donde:

- **192.168.1.200**: es la IP de nuestro servidor (tendrás que modificar y poner la IP de tu equipo).
- **dc**: es el nombre del servidor que hemos puesto durante la instalación (cuidado que sale dos veces).
- **midominio.com**: es el dominio que queremos configurar a continuación en Samba.

Con esto hecho, podremos continuar y configurar Samba como nuestro Controlador de Dominio.

5.5.2. Configurar Samba como Controlador de Dominio

Para comenzar con la configuración de Samba como un controlador de dominios para sistemas Windows tendremos que parar el servidor Samba:

Paramos todos los servicios de SAMBA

```
root@vega:~# smbcontrol all shutdown
```

Esto hace que los servicios de Samba (**smbd** y **nmbd**) se paren. Debido a que el fichero de configuración va a sufrir grandes cambios, y la configuración que tiene tras la instalación no es necesaria, se va a mover el fichero de configuración para guardarlo como backup:

Movemos el fichero de configuración original de SAMBA

```
root@vega:~# cd /etc/samba
root@vega:~# mv smb.conf smb.conf_backup
```

Al igual que hicimos con Windows Server a la hora de crear Active Directory, en Samba también vamos a necesitar de un DNS que esté gestionado por el propio controlador de dominio. En nuestro caso, como veremos más adelante, **Samba levantará un DNS propio**, pero tendremos que hacer que nuestro sistema GNU/Linux haga uso de él, tal como se explicará en el siguiente apartado.

Antes de eso, para realizar la configuración inicial del dominio, tenemos que ejecutar:

Usamos el asistente de creación del dominio

```
root@vega:~# samba-tool domain provision
```

Nos va a realizar una serie de preguntas, que tendremos que contestar:

- **Realm**: el nombre del Dominio que queremos crear. Ejemplo: **linuxdc.com**.
- **Domain**: Nombre de dominio **NETBIOS**. Debe ser una única palabra de máximo 15 caracteres. Por ejemplo linuxdc.
- **Server Role**: Es el tipo de rol que va a tener nuestro servidor. Puede ser uno de los siguientes:
  - **dc**: Domain Controller, es decir, controlador de dominio de Windows. Este es el modo que nos va a interesar en nuestro caso, ya que actuará como si se tratara de un Active Directory.

    En el fichero de configuración nos aparecerá como "active directory domain controller"
  - **member**: Miembro de un dominio. Nos permitiría que Samba sea un miembro de un dominio Windows.
  - **standalone**: Este es el modo por defecto una vez hemos instalado Samba, tal como se ha comentado previamente.
- **DNS backend**: Servidor backend de DNS. Al igual que Windows hacía uso de DNS, Samba cuando se convierte en controlador de dominio también. Existe la posibilidad de utilizar un servidor DNS propio (como Bind) pero en caso de usar "**SAMBA_INTERNAL**" Samba levantará un DNS propio.
- **DNS forwarder IP address**: El DNS al que se preguntará los DNS que desconocemos. Podemos poner el de Google, 8.8.8.8, o cualquier otro DNS público.
- **Administrator password**: Contraseña del administrador del Controlador de Dominio.

Una vez realizada la configuración inicial, antes de arrancar tenemos que instalar Winbind, que es necesario, y por si acaso lo paramos, para que después lo arranque Samba:

Paramos el servicio winbind

```
root@vega:~# systemctl stop winbind
```

Hay que recordar que **ahora mismo tenemos el servicio de Samba parado**, y lo único que hemos realizado ha sido la configuración inicial, por lo que el fichero de configuración `/etc/samba/smb.conf` ha sido modificado teniendo en cuenta las respuestas que hayamos dado previamente.

5.5.2.1. Arranque en modo interactivo

Tras realizar los pasos anteriores, nuestro servidor está listo para arrancar en modo Domain Controller, para ello en una consola ejecutaremos:

Arrancamos SAMBA en modo interactivo

```
root@vega:~# samba -F -i -d1
```

Los parámetros "-F -i -d1" sirve para arrancar Samba y que podamos ver qué está sucediendo durante la ejecución de Samba. Arrancarlo así nos sirve para poder ver si existe algún error en tiempo real. Una vez

todo sea correcto, podríamos parar la ejecución de este comando (con **ctrl+c**) y lanzar el demonio de manera normal.

Para ver los puertos que está usando Samba (como el 53 del DNS) podríamos ejecutar el siguiente comando:

Comprobamos los puertos utilizados por SAMBA

```
root@vega:~# ss -puntal | grep smb

Netid  State   Recv-Q  Send-Q   Local Address:Port  Peer Address:Port  Process
tcp    LISTEN  0       50       0.0.0.0:139         0.0.0.0:*     users:(("smbd",pid=6041,fd=47))
tcp    LISTEN  0       50       0.0.0.0:445         0.0.0.0:*     users:(("smbd",pid=6041,fd=46))
tcp    LISTEN  0       50       [::]:139            [::]:*        users:(("smbd",pid=6041,fd=45))
tcp    LISTEN  0       50       [::]:445            [::]:*        users:(("smbd",pid=6041,fd=44))
```

5.5.2.2. Usuarios en Samba Domain Controller

Al igual que sucedía en Active Directory, Samba tiene un sistema para la gestión y creación de usuarios que pertenecen al Domain Controller. A partir de ahora, la gran mayoría de los comandos que ejecutemos serán parámetros del siguiente comando, que es el sistema principal de administración de Samba:

Comando para controlar nuestro dominio en SAMBA

```
root@vega:~# samba-tool
```

Para crear los usuarios dentro del dominio gestionado por Samba podremos crearlos de la siguiente manera:

Crear nuevo usuario en SAMBA

```
root@vega:~# samba-tool user create profesor1
```

Si queremos visualizar todos los usuarios que existen en Samba podemos hacerlo con:

Crear nuevo usuario en SAMBA

```
root@vega:~# samba-tool user list
profesor1
Administrator
Guest
krbtgt
```

Tal como se puede ver, aparece el usuario “profesor1” creado previamente, junto con los usuarios que también se creaban por defecto en Active Directory:

- **Administrator**: administrador del dominio
- **Guest**: usuario invitado
- **krbtgt**: de manera resumida, usuario desactivado que es el encargado del sistema de autenticación (Kerberos).

Toda la gestión de usuarios del dominio Samba se podrá hacer con el comando `samba-tool user` añadiendo el correspondiente parámetro extra.

¡Atención!

No hay que confundir la gestión de usuarios de Samba con la propia de Linux, ya que son dos cosas completamente separadas.

5.5.2.3. Configuración del Controlador de Dominio

Tal como se ha comentado previamente, el comando samba-tool nos permitirá realizar todo tipo de configuraciones, como pueden ser:

- Gestión de ordenadores del dominio
- Gestión de usuarios
- Gestión de grupos
- …

5.5.3. Meter Windows 10 en Samba Domain Controller

Ha llegado el momento de meter un equipo Windows en el Dominio que tenemos gestionado por nuestro servidor Samba. Los pasos a dar serán los mismos que cuando metemos el equipo en un Active Directory.

- **Cambiar el nombre al equipo**: Esto hará que después sea más fácil realizar la gestión de equipos y saber a quién pertenece el equipo o dónde se sitúa.
- **Cambiar DNS** del equipo para que haga uso de la IP del servidor Samba.
- Vamos a "**Panel de Control > Sistema y Seguridad > Sistema**": Aquí le damos a "Cambiar configuración" dentro de la configuración de Dominio, y lo añadimos a nuestro Dominio creado previamente. Para añadirlo haremos uso del usuario Administrador que tenemos creado en el Controlador de Dominio Samba. Como pasaba al añadirlo a un Active Directory, el equipo tendrá que reiniciarse.

Una vez añadido el equipo, podremos loguearnos usando el usuario creado previamente.

5.6. Compartiendo carpetas mediante Samba

En algunas distribuciones, en la configuración por defecto de Samba ya aparece la compartición de ciertas carpetas. Normalmente suele aparecer la carpetas "home" de los usuarios, pero esta configuración es válida **cuando Samba está en modo "standalone"** siendo la configuración base la siguiente:

Ejemplo que aparece en /etc/samba/smb.conf

```
[homes]
    browseable = No
```

```
    comment = Home Directories
    create mask = 0700
    directory mask = 0700
    read only = No
    valid users = %S
```

Si desde un windows nos intentamos conectar a nuestro Samba haciendo uso de **\\IP_Samba\nombre_usuario** donde:

- **IP_Samba**: es la IP de nuestro servidor Samba
- **nombre_usuario**: es el usuario que tengamos en la base de datos de Samba veremos que se intentará conectar, y en caso de que coincida la contraseña de windows con la de Samba, podremos visualizar el directorio.

Información

Windows no tiene por qué estar dentro del dominio de Samba para acceder a una unidad compartida por Samba.

Esta configuración **no es válida cuando Samba está en modo controlador de dominio**, pero nos puede servir de base para crear nuestras carpetas compartidas.

5.6.1. Crear carpeta compartida

Para crear una carpeta compartida accesible por los usuarios creados en Samba, podemos modificar el fichero de configuración y añadir una sección que sea similar a la siguiente:

Configuración para compartir una carpeta en SAMBA

```
[sistemas]
    browseable = No
    comment = Carpeta Sistemas compartida
    path = /home/sistemas
    create mask = 0777
    directory mask = 0777
    read only = No
    valid users = profesor1
```

Esta sección cuenta con distintos parámetros que hay que entender:

- **[sistemas]**: es el nombre de la carpeta tal como accederemos desde Windows.
- **browseable**: indica si la carpeta puede ser “buscada” accediendo al servidor o si debemos conocer cómo se llama la carpeta para acceder a ella directamente.

- **comment**: comentario sobre la carpeta compartida.
- **path**: ruta en el servidor GNU/Linux donde realmente se encuentra la carpeta compartida.
- **create mask**: cuando se crea un fichero en la carpeta compartida, los permisos que tendrá el fichero en GNU/Linux.
- **directory mask**: cuando se crea un directorio en la carpeta compartida, los permisos que tendrá el fichero en GNU/Linux.
- **read only**: si queremos que la carpeta compartida sea de modo sólo lectura.
- **valid users**: usuarios que podrán acceder a la carpeta compartida. Estos usuarios deberán haber sido creados previamente en Samba (mirar apartado de creación de usuarios).
    - Si queremos que puedan conectarse varios usuarios, deberemos poner el listado separado por espacios.
    - Si queremos que puedan conectarse a esta carpeta todos los usuarios de un grupo, deberemos poner **@nombre_grupo** en esta variable.

5.7. Administración remota de Samba, desde Windows

Para facilitar la administración del **Controlador de Dominio Samba** que acabamos de crear, podemos hacer uso de las "**Herramientas de administración remota del servidor**" desde un equipo Windows.

Desde las actualizaciones de octubre de 2018 podemos hacer la instalación de las herramientas **RSAT** (*Remote Server Administration Tools*) a través de las "**Características opcionales**". Si no, deberíamos instalarlas esta actualización.

Para poder usar las herramientas que instalemos tenemos que estar logueados en Windows con un usuario administrador del dominio. Las herramientas instaladas están en "*Panel de Control → Sistema y Seguridad → Herramientas Administrativas*".

6. Creación de copias de seguridad en GNU/Linux

Tal como hemos visto, la gestión de copias de seguridad es de vital importancia para preservar los datos de nuestra empresa. Debemos interiorizar que el realizar copias de seguridad, tanto en nuestro ámbito personal como profesional, es una parte en la gestión de nuestra información así como el comprobar que se realizan de manera correcta.

Al igual que sucede en Windows, existen múltiples sistemas para realizar copias de seguridad en GNU/Linux, pero uno de los más sencillos de utilizar ya suele venir instalado en la gran mayoría de las distribuciones hoy día: **rsync**.

6.1. Rsync como sistema para sincronizar directorios

Rsync es una aplicación que ofrece la posibilidad de realizar la sincronización de directorios (de manera local o en remoto) de manera eficiente, ya que es capaz de sincronizar sólo las modificaciones realizadas en los ficheros.

Por defecto, **rsync** copia las modificaciones o ficheros nuevos que existen en el directorio de origen y los copia al directorio de destino. Si un fichero es borrado en el directorio origen, **por defecto no se borra en el directorio de destino**.

La manera más sencilla de utilizar rsync es para crear una sincronización de un directorio local a un servidor remoto en el que almacenar una copia de los datos.

¡Cuidado!

Esta sincronización remota equivale a realizar una copia completa de todos los datos (de manera recursiva), o lo que es lo mismo, un *full-backup*.

Ejemplo para sincronizar dos directorios locales

```
ruben@vega:~$ rsync -av directorio_origen  directorio_destino
```

En el ejemplo anterior lo que hace es sincronizar todo el "directorio_origen" dentro del "directorio_destino". En cambio:

Ejemplo para sincronizar dos directorios locales

```
ruben@vega:~$ rsync -av directorio_origen/  directorio_destino2
```

En este ejemplo, como hemos añadido una barra "/" al final del directorio origen, lo que estamos indicando es que el contenido (y sólo el contenido) se va a sincronizar dentro de "directorio_destino2".

¡Cuidado!

Hay que tener cuidado con esa "**/**", ya que ponerla o no ponerla da resultados distintos en la sincronización.

6.1.1. Rsync para sincronizar de manera remota

Tal como sabemos, nunca deberíamos realizar copias de seguridad de manera local en el mismo disco duro, ni tenerlas en el mismo equipo, por lo que es necesario poder realizar copias de seguridad desde otros servidores o a otros servidores.

Imaginemos que queremos copiar los datos de una carpeta compartida `/home/sistemas` al servidor remoto 10.40.30.200, y dejar una copia en `/home/backups`:

Sincronizar directorio local a servidor remoto

```
ruben@vega:~$ rsync -av /home/sistemas  root@10.40.30.200:/home/backups
```

Con este comando realizaremos la sincronización al servidor remoto haciendo uso de SSH y conectándonos con el usuario root remoto. Explicación de los distintos apartados del comando:

- **rsync** es el programa que usamos para realizar la sincronización.
- **-av** son los parámetros que realizan la sincronización en moodo archivos y en modo "verbose" (nos indica qué está sincronizando).
- **/home/sistemas** es el directorio con los datos de origen que vamos a sincronizar
- **root** es el usuario con el que nos vamos a conectar al servidor remoto.
- **@10.40.30.200:** es el servidor al que nos vamos a conectar por SSH para realizar en él la copia. **¡OJO!** Es importante esa "@" y los ":" al final.
- **/home/backups** es el lugar donde dejaremos una copia de los datos a sincronizar.

Tal como se ha comentado, rsync hará uso de SSH para realizar la conexión remota, por lo que los datos se enviarán de manera segura y cifrada.

6.1.2. Obtener datos remotos

Si queremos realizar la sincronización en el orden inverso, obtener los datos estando en el servidor de backup y traernos los datos de un servidor remoto, el comando sería:

Sincronizar directorio remoto a directorio local

```
root@backups:~# rsync -av 10.40.30.5:/home/sistemas  /home/backups
```

En este caso, estando en el servidor de backups, vamos a traernos los datos del servidor 10.40.30.5. En este caso, antes de la IP del servidor no hemos puesto usuario, por lo que se realizará la conexión con el mismo usuario que somos actualmente, en este caso **root**.

6.1.3. Opciones extra

Hasta ahora hemos visto las opciones más básicas para sincronizar directorios (tanto locales como remotos), pero rsync cuenta con muchas opciones extra que es interesante conocer:

- **-z** o **--compress** sirve para comprimir los datos antes de realizar el envío o recepción de los datos. Sólo es útil si vamos a realizar la sincronización de manera remota.
- **--progress** sirve para ver el progreso de cómo va la transferencia de los ficheros
- **--delete** sirve para borrar en el destino los ficheros que no existen en el origen.
- **--exclude "*txt"** excluye los ficheros que termina con la extensión "txt" al hacer la sincronización.

7. Comandos de administración básica en GNU/Linux

En este documento vamos a recopilar comandos que nos pueden ser útiles a la hora de usar un sistema GNU/Linux y realizar su administración.

7.1. Comandos sobre el sistema de ficheros

A continuación unos comandos básicos para utilizar sobre el sistema de ficheros.

Listar el contenido del directorio donde nos encontramos

```
ruben@vega:~$ ls
Descargas   Escritorio  Música      Público
Documentos  Imágenes    Plantillas  Vídeos
```

Listar el directorio actual, versión “long”

```
ruben@vega:~$ ls -l
total 36K
drwxr-xr-x 2 ruben ruben 4,0K nov 12  2022 Descargas
drwxr-xr-x 2 ruben ruben 4,0K dic  6 09:50 directorio1
drwxr-xr-x 2 ruben ruben 4,0K nov 12  2022 Documentos
drwxr-xr-x 2 ruben ruben 4,0K nov 12  2022 Escritorio
drwxr-xr-x 2 ruben ruben 4,0K nov 12  2022 Imágenes
drwxr-xr-x 2 ruben ruben 4,0K nov 12  2022 Música
drwxr-xr-x 2 ruben ruben 4,0K nov 12  2022 Plantillas
drwxr-xr-x 2 ruben ruben 4,0K nov 12  2022 Público
drwxr-xr-x 2 ruben ruben 4,0K nov 12  2022 Vídeos
```

Crear un nuevo directorio

```
ruben@vega:~$ mkdir directorio1
ruben@vega:~$ ls
Descargas    Documentos  Imágenes  Plantillas  Vídeos
directorio1  Escritorio  Música    Público
```

Borrar un directorio que está vacío

```
ruben@vega:~$ rmdir directorio1
```

Editar un fichero

```
ruben@vega:~$ nano fichero.txt
```

Obtener el contenido de un fichero de texto

```
ruben@vega:~$ cat fichero.txt
hola, qué tal
```

Paginar el contenido de un fichero de texto

```
ruben@vega:~$ more fichero.txt
hola, qué tal
```

Borrar un fichero

```
ruben@vega:~$ rm fichero.txt
```

Borrar un directorio y todo el contenido que tiene dentro. ¿Diferencias?

```
ruben@vega:~$ rm -r directorio2
ruben@vega:~$ rm -fr directorio2
```

Buscar por un contenido dentro de un fichero

```
ruben@vega:~$ grep hola fichero.txt
```

7.2. Comandos de red

Para ver los interfaces de red y las direcciones IP que tienen

Obtener los interfaces y las IPs

```
ruben@vega:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo

2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
link/ether 1a:8a:1c:ff:25:15 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.99/24 brd 192.168.1.255 scope global enp4s0
```

Para ver la ruta por defecto (el gateway o puerta de enlace).

Obtener la puerta de enlace

```
ruben@vega:~$ ip route show default
default via 192.168.1.1 dev enp4s0 onlink
```

Ver los puertos TCP y servicios que están a la escucha en nuestro servidor

Listar los puertos TCP a la escucha

```
root@vega:~# ss -pntal
```

7.3. Comandos sobre procesos

Listar todos los procesos

Listar todos los procesos

```
root@vega:~# ps aux
```

Listar todos los procesos en forma de árbol (para saber de quién son hijos)

Listar todos los procesos en forma de árbol

```
root@vega:~# pstree -p
```

Matar un proceso (donde PID es el identificador del proceso).

Matar un proceso

```
root@vega:~# kill -9 PID
```

7.4. Estado de la carga y memoria del servidor

Para ver los procesos y su estado por consumo de CPU, RAM…

Ver el estado del servidor

```
root@vega:~# top
```

Para ver los procesos y su estado por consumo de CPU, RAM… es necesario instalar este paquete

Ver el estado del servidor

```
root@vega:~# htop
```

7.5. Comandos sobre servicios (systemd/systemctl)

GNU/Linux cuenta con un sistema unificado (**systemd**) para administrar el sistema y los servicios que tenemos en nuestro servidor. Dado que es una pieza fundamental en el sistema operativo, debemos de conocer ciertos comandos para poder desempeñar tareas con él.

Listar todos los servicios/unidades

Listar todos los servicios

```
root@vega:~# systemctl
```

Comprobar si algún servicio ha fallado

Comprobar servicios que han fallado

```
root@vega:~# systemctl --failed
```

Comprobar el estado de un servicio concreto (en este caso, ssh)

Comprobar servicios que han fallado

```
root@vega:~# systemctl status ssh
```

Parar un servicio concreto

Parar un servicio concreto

```
root@vega:~# systemctl stop ssh
```

Arrancar un servicio concreto

Arrancar un servicio concreto

```
root@vega:~# systemctl start ssh
```

Ver los logs de todo el sistema

Ver los logs del sistema

```
root@vega:~# journalctl
```

Ver los logs de un servicio concreto (en este caso, ssh)

Ver los logs del sistema

```
root@vega:~# journalctl -u ssh
```

Ver los logs del kernel

Ver los logs del kernel

```
root@vega:~# journalctl -k
```

7.6. Comandos para instalar/desinstalar paquetes

Los comandos que se van a exponer aquí sirven para las distribuciones que utilizan el sistema de paquetes APT (Advanced Packaging Tool) que está presente en las distribuciones derivadas de Debian, como por ejemplo Ubuntu.

Sincroniza el índice de paquetes local con lo que está en los repositorios remotos configurados en /etc/apt/sources.list

```
root@vega:~# apt update
```

Instala las versiones más nuevas de los paquetes que ya tenemos instalados

```
root@vega:~# apt upgrade
```

Instala las versiones más nuevas de los paquetes teniendo en cuenta las dependencias de los mismos y solucionando posibles conflictos.

```
root@vega:~# apt full-upgrade
```

7.7. Comandos para apagar/reiniciar

Para realizar el apagado o reinicio de un equipo se pueden utilizar varios comandos distintos.

Reiniciar el equipo

```
root@vega:~# reboot
```

Apagar el equipo en un minuto

```
root@vega:~# shutdown
```

Apagar el equipo ahora

```
root@vega:~# shutdown now
```

Suspender el equipo

```
root@vega:~# systemctl suspend
```

III

Anexos

1. Instalar Ubuntu 22.04 LTS

En este anexo realizaremos la instalación de la distribución Ubuntu 22.04 LTS en su versión para servidores. En este anexo no se va a explicar cómo realizar la creación de una máquina virtual donde se aloja el sistema operativo, ya que existen distintos tipos de virtualizadores.

No se realizará una guía "paso a paso", sino que se centrará en las partes más importantes de la instalación y en las que más dudas puedan surgir.

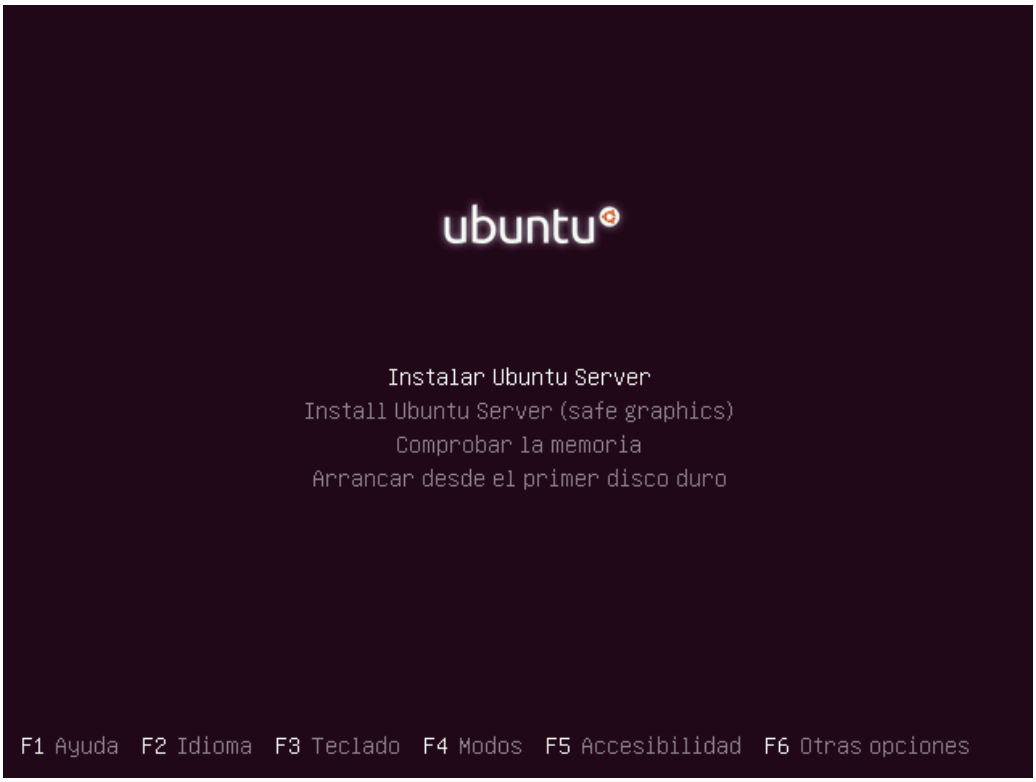
1.1. Descargar Ubuntu 22.04

La ISO la obtendremos de la web oficial y seleccionaremos la versión 22.04 LTS de Ubuntu Server. Esta ISO contendrá el sistema base de Ubuntu y nos guiará para realizar la instalación del sistema operativo.

Una vez descargada la ISO tendremos que cargarla en el sistema de virtualización elegido y arrancar la máquina virtual.

1.2. Instalar Ubuntu 22.04

Tras arrancar la máquina virtual nos aparecerá un menú para seleccionar el idioma durante la instalación y le daremos a "Instalar Ubuntu Server".



A partir de aquí comenzará el instalador y los pasos que nos aparecerán serán los siguientes (algunos de estos pasos puede que no estén 100 % traducidos al castellano):

1. Elegir el idioma del sistema
2. Actualización del instalador:

- Si la máquina virtual se puede conectar a internet, comprobará si existe una actualización del propio instalador de Ubuntu.
- Podemos darle a “Continuar sin actualizar”

3. Configuración del idioma del teclado
4. Configuración de la red
5. Configuración del proxy de red
6. Configuración del “mirror” o servidor espejo desde donde descargarse los paquetes de software para las actualizaciones posteriores.
7. Selección del disco duro donde realizar la instalación
8. Elegir el particionado de disco.
9. Configuración del perfil. Introduciremos el nombre de usuario, el nombre del servidor y la contraseña del usuario que vamos a crear.
10. Configuración de SSH Server. Aceptaremos que se instale el servidor SSH durante la instalación. En caso de no seleccionar esta opción, posteriormente podremos realizar la instalación.
11. “Featured Server Snaps”. En esta pantalla nos permite instalar software muy popular en servidores.

Una vez le demos a continuar, comenzará la instalación en el disco duro. Debido a que durante la instalación tenemos conexión a internet, el propio instalador se descarga las últimas versiones de los paquetes de software desde los repositorios oficiales.

Al terminar la instalación, tendremos que reiniciar la máquina virtual.

1.3. Post-instalación

Tras realizar el reinicio de la máquina virtual nos encontraremos con que el sistema arranca en el sistema recién instalado, y que tendremos que loguearnos introduciendo el usuario y la contraseña utilizadas en la instalación.

1.3.1. Actualización del sistema

Por si acaso, realizaremos la actualización del índice del repositorio, actualizaremos el sistema y en caso necesario realizaremos un nuevo reinicio:

Actualizar Ubuntu

```
root@ubuntu:~# apt update
...
root@ubuntu:~# apt upgrade
...
```

Con estos comandos nos aseguramos que el sistema está actualizado a los últimos paquetes que están en el repositorio.

1.3.2. Poner IP estática

Debido a la configuración de red de nuestro servidor, la IP está puesta en modo dinámica, esto quiere decir que nuestro equipo ha cogido la IP por configuración de DHCP de nuestra red. Debido a que un servidor debe de tener IP estática, tenemos que realizar la modificación adecuada para ponerle la IP estática que mejor nos convenga. Para ello editaremos el fichero de configuración situado en la siguiente ruta: `/etc/netplan/00-installer-config.yaml`

Lo modificaremos para que sea parecido a (siempre teniendo en cuenta la IP y gateway de nuestra red):

Configurando IP estática en Ubuntu

```
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [192.168.1.199/24]
      routes:
      - to: default
        via: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8,1.1.1.1]
  version: 2
```

El fichero de configuración que hemos modificado es de tipo YAML, que es un formato de texto que suele ser utilizado en programación o en ficheros de configuración. Este tipo de ficheros tiene en cuenta los espacios para el uso de la identación, y no suele permitir el uso de tabuladores.

Para aplicar los cambios realizados en el fichero de configuración deberemos ejecutar el siguiente comando que aplicará los cambios:

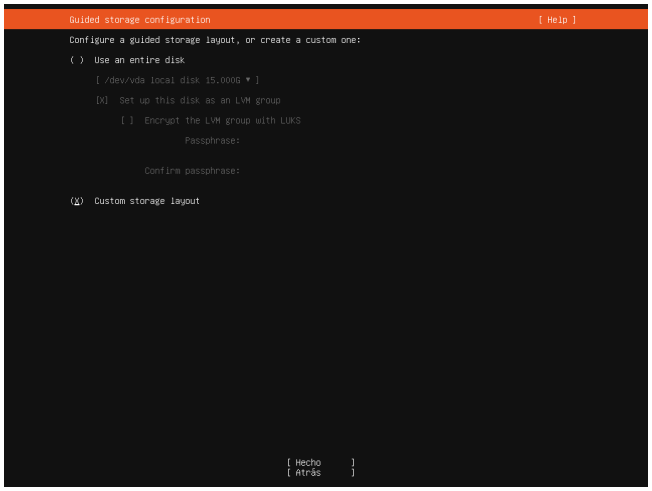
Aplicar configuración de IP

```
root@ubuntu:~# netplan apply
```

2. Configurar RAID 1 durante la instalación de Ubuntu

Tal como hemos podido ver anteriormente, durante la instalación de Ubuntu 20.04, en el paso 7 podemos realizar la instalación en el disco duro que tengamos instalado en el servidor físico o en la máquina virtual.

En este paso podemos realizar distintas configuraciones:



- Usar el disco entero.
    - Nos permitirá crear un sistema con LVM (por defecto activado) y con posibilidad de cifrar la partición creada.
- Crear un diseño de almacenamiento personalizado.

En la segunda opción podremos:

- Crear particiones a nuestro gusto.
- Elegir el sistema de ficheros de las particiones.
- Crear sistema RAID por software.

2.1. Pasos previos

Dado que vamos a crear un sistema RAID 1 durante la instalación de Ubuntu, necesitaremos al menos **dos discos duros** en nuestro servidor antes de comenzar con la instalación.

En nuestro sistema virtualizado hemos añadido dos discos duros de igual tamaño (15GB), en los cuales crearemos particiones para posteriormente sobre ellas realizar el RAID 1.

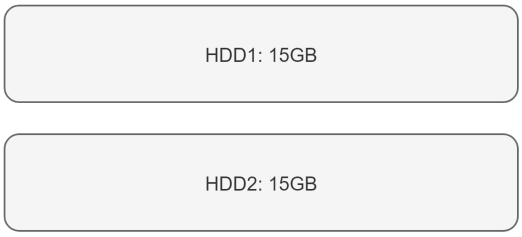
2.2. Entendiendo las particiones a realizar

En este apartado vamos a explicar la teoría que está detrás del sistema de particionado que vamos a necesitar crear y que posteriormente realizaremos en el sistema de instalación de Ubuntu.

2.2.1. Situación inicial: discos duros sin particionar

Como ya se ha comentado, en nuestro servidor vamos a contar con dos discos duros de igual tamaño. Esto suele ser lo habitual, pero lo importante es que las particiones que vayamos a crear sean del tamaño exacto, aunque un disco duro sea de mayor tamaño (aunque lógicamente, ese espacio quedará desaprovechado).

En la siguiente imagen vemos que tenemos dos discos duros de 15GB de tamaño cada uno:



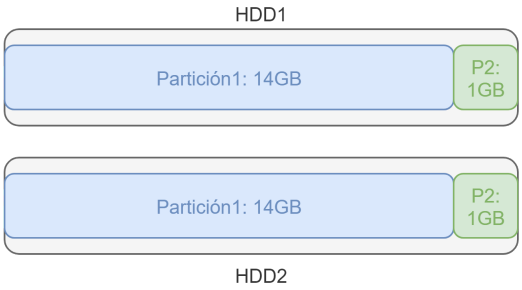
2.2.2. Particionado inicial

A continuación vamos a tener que pensar cómo van a ser las particiones que vamos a necesitar en nuestro servidor. En nuestro caso vamos a crear dos:

- **14GB**: Sistema operativo.
- **1GB**: (o hasta completar) SWAP.

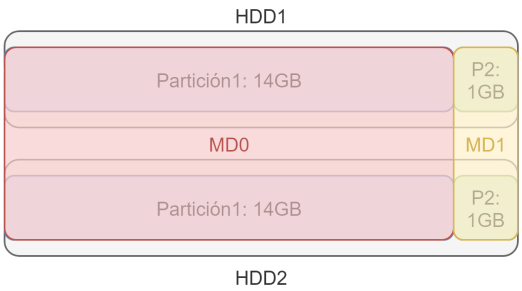
Como se puede entender, al tener una única partición, todo el sistema raiz "/" va a ir en ella, mientras que la otra partición será la usada para el área de intercambio.

Es importante entender que en este paso sólo **vamos a crear las particiones pero sin darles formato**. Por lo tanto, nuestros discos duros ahora tendrían este aspecto:



2.2.3. Crear particiones RAID

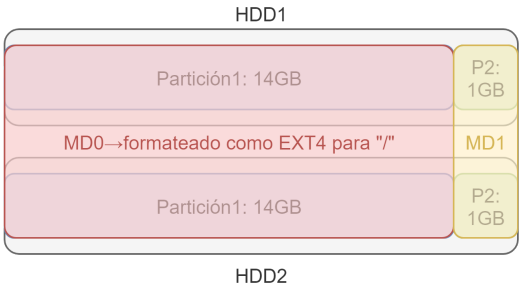
El siguiente paso es crear las particiones "virtuales" RAID. Vamos a crear una primera partición RAID que va a incluir las particiones de 14GB de ambos discos duros, y la segunda partición virtual incluirá las particiones de 1GB.



De esta manera, tendremos unas particiones MD0 y MD1 que son particiones virtuales.

2.2.4. Formatear particiones RAID con el formato adecuado

El último paso de la instalación es hacer uso de las particiones RAID creadas y formatearlas con el sistema de ficheros acorde a las necesidades que tengamos, y elegir el punto de montaje adecuado.



En nuestro caso va a ser:

- **MD0**: sistema de ficheros ext4 y lo vamos a utilizar como sistema de ficheros raíz "/".
- **MD1**: formateado como SWAP y actuará como área de intercambio.

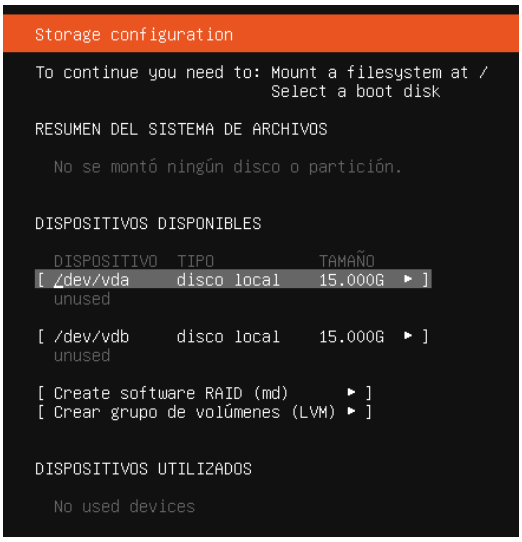
Tras este paso, la instalación del sistema operativo puede continuar de la manera habitual.

2.3. Realizando el particionado en el instalador de Ubuntu

Tras haber entendido las particiones que vamos a realizar, ahora es el momento de proceder en el instalador de Ubuntu. Vamos a seguir los mismos pasos que hemos explicado en el apartado anterior, de esta manera aplicaremos lo aprendido a nivel teórico.

2.3.1. Situación inicial: discos duros sin particionar

Tal como hemos comentado previamente, en el paso 7 de la instalación, elegiremos la opción de crear un "diseño de almacenamiento personalizado". Tras entrar en esta opción, el instalador tendrá el siguiente aspecto:



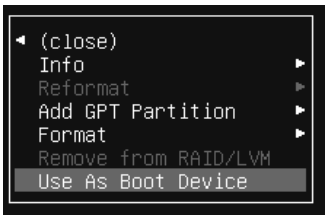
Tal como se puede ver, tenemos dos discos duros en el sistema: **vda** y **vdb**. El nombre de los discos viene de **V**irtual **D**isk, dado que la instalación la estamos realizando en una máquina virtual.

2.3.2. Particionado inicial

En este paso vamos a crear en cada uno de ellos la partición de 14GB y el resto del espacio la usaremos para la segunda partición.

2.3.2.1. Marcar discos como dispositivos de arranque

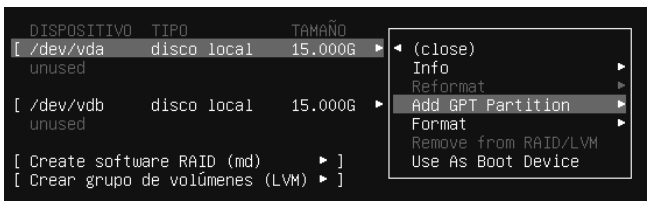
Por cómo funciona el sistema de arranque de Linux, antes de realizar las particiones vamos a marcar que ambos discos duros sean dispositivos de arranque ("Boot Device"). Para ello pulsaremos "Intro" en cada uno de los discos y elegiremos la opción correspondiente (imágenes de cada disco):



De esta manera, el instalador de Ubuntu creará una pequeña partición al inicio del disco donde al terminar se realizará la instalación del sistema de arranque GRUB en ambos discos duros.

2.3.2.2. Crear particiones

Ahora es el momento de crear las particiones, y los pasos serán seleccionar el disco duro, pulsar "Intro", se nos desplegará un pequeño menú y vamos a elegir la opción *"Add GPT Partition"* y rellenaremos el tamaño de la partición que nos interese en el momento y el formato lo dejaremos en *"Leave unformatted"* (dejar sin formatear).



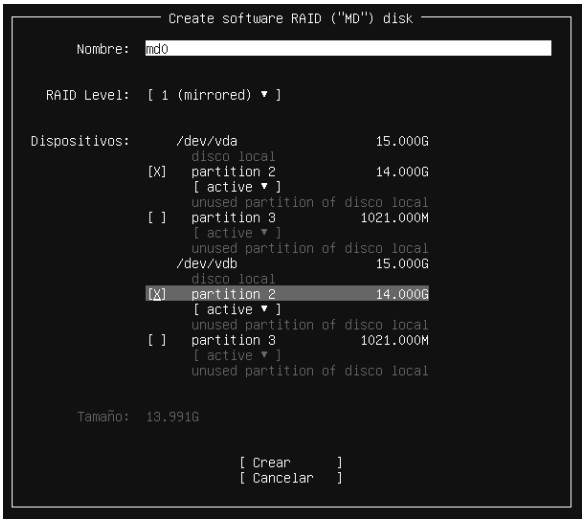
Estos pasos lo realizaremos en cada disco duro con las particiones que vamos a necesitar, quedando al finalizar el sistema así:

Tal como se puede ver, cada disco duro tiene dos particiones con el tamaño deseado que no están siendo utilizadas, y en la parte de abajo aparecen las particiones denominadas “BIOS grub spacer”.

2.3.3. Crear particiones RAID

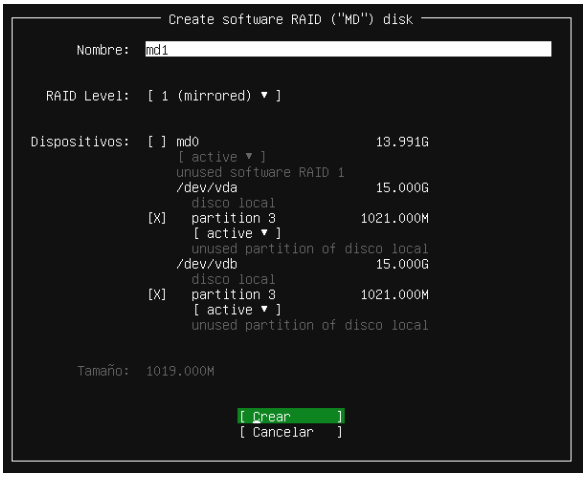
El siguiente paso es crear las particiones RAID en las que haremos que el sistema cree un RAID 1 haciendo uso de las particiones de los discos duros físicos. Seleccionaremos la opción “*Create software RAID (md)*” y nos aparecerá una ventana en la que podremos elegir:

- **Nombre**: de la partición que vamos a crear. Es habitual que estas particiones empiecen por “**md**”, ya que viene de “*multiple device*”.
- **Nivel RAID**: Podremos elegir entre las versiones 0, 1, 5, 6 y 10 de RAID. Por defecto está seleccionada la opción RAID 1.
- **Dispositivos**: sobre el que aplicaremos el RAID.

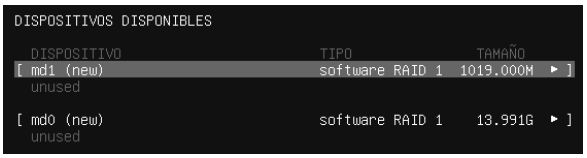


Crearemos primero **md0** seleccionando las particiones de 14GB tal como aparece en la siguiente imagen:

Y a continuación crearemos **md1** con las particiones restantes. Tal como se puede ver a continuación, las particiones de 14GB ya no aparecen, porque están siendo usadas en el otro RAID.



Tras la creación de los dispositivos “md”, nos aparecerán como dispositivos disponibles para usarlas en el siguiente paso:



2.3.4. Formatear particiones RAID con el formato adecuado

Aunque este paso lo vamos a realizar sobre las particiones RAID creadas previamente, es un paso que es habitual realizar cuando queremos que nuestra instalación tenga un sistema de particiones propio.

Tenemos que pensar que las particiones RAID ahora son como particiones normales, a las que les vamos a querer dar un formato y utilizarlas para realizar la instalación.

Vamos a seleccionar la primera, **md0**, posicionándonos encima de ella y dándole a “Intro” y posteriormente dándole a “*Add GPT partition*”:

Dejaremos el tamaño en blanco, indicando que usaremos todo el espacio libre, lo vamos a formatear con el sistema de ficheros **ext4** y se va a montar como el sistema de ficheros “/”.

En **md1** también le daremos a crear nueva partición, y haremos lo mismo pero usando el formato **swap**. Quedando una vez terminado el sistema de particiones de la siguiente manera:

Y tras esto podremos continuar la instalación de manera normal.

3. Administración remota

En informática no siempre tenemos los equipos que administramos en nuestra oficina. Pueden estar en otro edificio, la oficina de un cliente, en internet … por lo tanto no siempre es posible acceder de manera física a ellos, y por tanto entra en juego la **administración remota**.

Podemos definir la administración remota como el sistema que nos permite realizar ciertas acciones "lanzadas" desde nuestro equipo local pero que serán ejecutadas en un equipo remoto.

Se pueden diferenciar varios tipos de sistemas dentro de la administración remota, pero nos vamos a centrar en los siguientes:

- **Cliente remoto**: Lanzamos una acción a ejecutar desde un equipo remoto a través de algún tipo de interfaz o comando (que viajará a través de un protocolo securizado) y esperaremos a la respuesta.
- **Acceso remoto**: En este caso lo que hacemos es conectarnos al equipo a través de un protocolo que nos va a permitir administrarlo como si estuviésemos delante de él.

Todos estos sistemas pueden ser complementarios, y puede que podamos administrar un mismo servicio de todas estas maneras, por lo que queda a nuestra disposición elegir el mejor método en cada momento.

Por otro lado, dependiendo de qué tipo de administración vayamos a llevar a cabo, o el protocolo que utilice, tendremos que tener acceso al servidor de alguna manera (ya sea conexión directa o mediante VPN).

Información

Dependiendo de la administración remota que realicemos, necesitaremos conexión directa o mediante VPN al equipo que nos queremos conectar.

Por último, también debemos de conocer el tipo de protocolo que vamos a utilizar al realizar la conexión remota y por dónde va a pasar esa comunicación. Siempre hay que premiar la seguridad de la comunicación, y más cuando esta puede pasar por redes no controladas. Por lo tanto, deberemos asegurar que el protocolo utilizado es seguro, o en caso contrario, securizarlo de alguna manera.

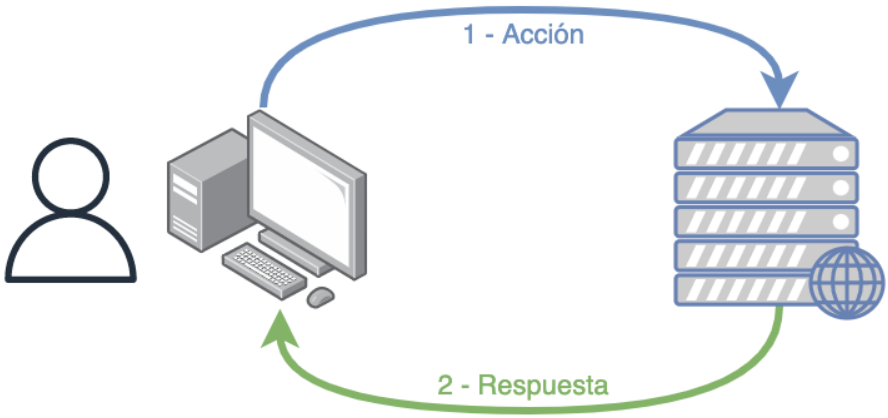
¡Cuidado!

Siempre debemos confirmar que la comunicación que se realiza para la administración remota viaja cifrada.

Más adelante veremos cómo securizar una comunicación no segura realizando un túnel mediante el protocolo SSH en entornos GNU/Linux.

3.1. Cliente remoto

Este sistema de administración permite enviar acciones al equipo remoto a través de un protocolo establecido, y dependiendo de la acción ejecutada se esperará una respuesta o no.



Hoy en día es muy habitual este tipo de sistemas a través de disintos **CLI** (*client line interface*) o **GUI** (*graphic user interface*) que nos permiten administrar servicios remotos. Por ejemplo:

- **https://www.mysql.com/MySQL**: El sistema gestor de base de datos MySQL cuenta con un cliente para realizar la conexión, ya sea desde el propio equipo o desde un equipo remoto.

  Este cliente se puede ejecutar desde línea de comandos, aunque también viene integrado en distintos interfaces gráficos como MySQL Benchmark, Dbeaver, ...

- **AWS CLI**: Es el interfaz de línea de comandos para poder administrar de manera remota los servicios contratados en la nube AWS de Amazon.

- **Gcloud CLI**: Similar al caso anterior pero esta vez para Google Cloud.

- **Remote Server Administration Tools for Windows 10**: En este caso se trata de un interfaz gráfico (**GUI**) que nos permite administrar un Windows Server desde un equipo Windows 10.

Antes de poder realizar la conexión remota con el cliente **debemos configurar un sistema de autenticación** para que el servicio remoto acepte las peticiones enviadas. En algunos casos será usando unos sistemas de certificados y en otros introduciendo un usuario y contraseña que establecerá una sesión temporal.

En el caso de AWScli y GCloud no nos estamos conectando directamente a nuestros servidores alojados en esas nubes, si no que lanzamos la orden a un "proxy" que verificará nuestros credenciales, verá los permisos que tenemos y después realizará la acción solicitada.

3.2. Acceso remoto

Este sistema permite acceder al sistema y podremos administrarlo como si nos encontrásemos delante. Dependiendo del sistema la conexión nos permitirá interactuar de alguna de las siguientes maneras:

- **CLI**: Mediante una conexión de línea de comandos. Es el caso más habitual en servidores GNU/Linux y la conexión se hace a través del protocolo seguro **SSH**.

- **GUI**: Podremos obtener un interfaz gráfico con el que veremos lo que está ocurriendo en pantalla en ese momento. En este caso, dependiendo del sistema, existirán distintas opciones, pero vamos a nombrar dos de ellas:

- **RDP**: Es el protocolo de escritorio remoto de Microsoft que transmite la información gráfica que el usuario debería ver por la pantalla, la transforma en el formato propio del protocolo, y la envía al cliente conectado. El problema es que este sistema desconecta al usuario que está logueado para poder hacer uso del escritorio remoto.
- **VNC**: En inglés *Virtual Network Computing*, es un servicio con estructura **cliente-servidor** que permite visualizar el escritorio del servidor desde un programa cliente. En este caso, no existe desconexión del usuario que está logueado y por tanto podrá ver lo que le están realizando de manera remota.

  Es muy habitual que los equipos de usuarios ya tengan la instalación del servidor hecha, para que de esta manera, en caso de incidencia, poder realizar la conexión remota sin que el usuario tenga que realizar ninguna acción.

A continuación se va a detallar algunos de los métodos mencionados.

3.3. SSH

SSH es un protocolo de comunicación segura mediante cifrado cuya función principal es el acceso remoto a un servidor. La arquitectura que utiliza es la de cliente-servidor.

Aunque el uso más habitual de SSH es el acceso remoto, también se puede utilizar para:

- Securizar protocolos no seguros mediante la realización de túneles.
- Acceder a un equipo saltando a través de otro.

Estas funcionalidades las veremos más adelante.

3.3.1. Servidor SSH

En el servidor al que nos queramos conectar deberá estar instalado el demonio/servicio SSH, conocido como **sshd**. Es habitual que ya esté instalado en sistemas GNU/Linux, pero de no ser así deberemos usar el sistema de paquetes de nuestra distribución para hacer la instalación. El nombre suele ser **openssh-server**.

Este servicio por defecto se pondrá a la escucha en el puerto 22/TCP:

SSHd escuchando en puerto 22

```
ruben@vega:~$ sudo ss -pntaln
State   Recv-Q   Send-Q   Local Address:Port   Peer Address:Port   Process
LISTEN  0        128      0.0.0.0:22           0.0.0.0:*           users:(("sshd",pid=1122,fd=3))
LISTEN  0        128      [::]:22              [::]:*              users:(("sshd",pid=1122,fd=4))
```

La configuración del servicio se realiza a través de un fichero de configuración que está situado en la ruta `/etc/ssh/sshd\_config`. Las distribuciones de GNU/Linux ya traen una configuración predeterminada que suele constar de las siguientes directivas (aunque hay muchas más):

- **Port**: Normalmente viene comentada, ya que el puerto por defecto es el 22. En caso de querer cambiar el puerto, podremos modificar esta línea, asegurando que no esté comentada.

- **ListenAddress**: Por defecto SSH se pondrá a la escucha en todos los interfaces que tengamos configurados. Si sólo nos interesa escuchar en alguna de las IPs que tengamos configuradas, deberemos modificar esta configuración.
- **PermitRootLogin**: Para evitar problemas de seguridad, esta directiva suele estar configurada a "**No**", para evitar que se puedan usar los credenciales de root para hacer el login.
- **PubkeyAuthentication**: Esta directiva permite realizar la conexión a través de unas claves públicas/privadas que podemos crear. Se explicará más adelante.

Hoy día también se puede instalar en Windows 10 y posteriores a través de un comando, siendo administrador de PowerShell:

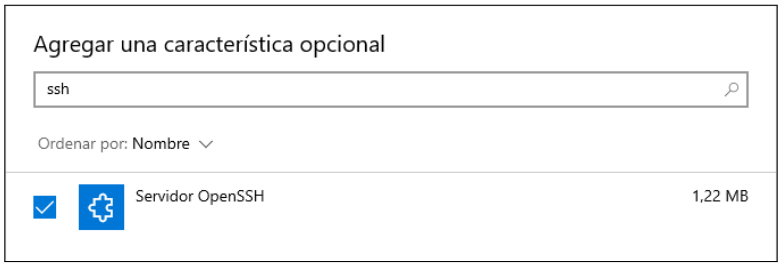
Instalando OpenSSH Server en Windows 10

```
PS C:\Windows\System32> Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0

PS C:\Windows\System32> Start-Service sshd

PS C:\Windows\System32> Set-Service -Name sshd -StartupType 'Automatic'
```

O desde el interfaz gráfico a través de las "**Características opcionales**", buscando por ssh:



3.3.2. Cliente SSH

El cliente SSH es aquel programa que a través del protocolo SSH se puede conectar a un servidor SSH. Existen distintos tipos de clientes que podemos utilizar:

- **CLI**: El cliente de consola es el más habitual. Está instalado de forma habitual en todas las distribuciones de GNU/Linux (normalmente el paquete se llama **openssh-client**). También lo encontramos instalado por defecto en MacOS.

Hoy día también está instalado en Windows 10, y de no estar, se puede instalar a través de las "**Características Opcionales**".

- **GUI**: Existen distintos interfaces gráficos que nos puede interesar utilizar:
  - **Putty**: Un cliente muy habitual en entornos Windows.
  - **Kitty**: Una versión mejorada del anterior.
  - **Termius**: Cuenta con versión de escritorio y móvil.

Para realizar la conexión al servidor SSH debemos conocer:

- **Dirección del servidor**: Ya sea mediante IP o nombre FQDN (*fully qualified domain name*) que se resuelva.
- **Puerto**: Ya hemos comentado que por defecto el puerto es 22.
- **Usuario**: Para realizar el sistema de autenticación, necesitamos un usuario que exista en el sistema.
- **Contraseña**: Los credenciales de acceso del usuario.

Para realizar la conexión desde un cliente de consola ejecutaremos:

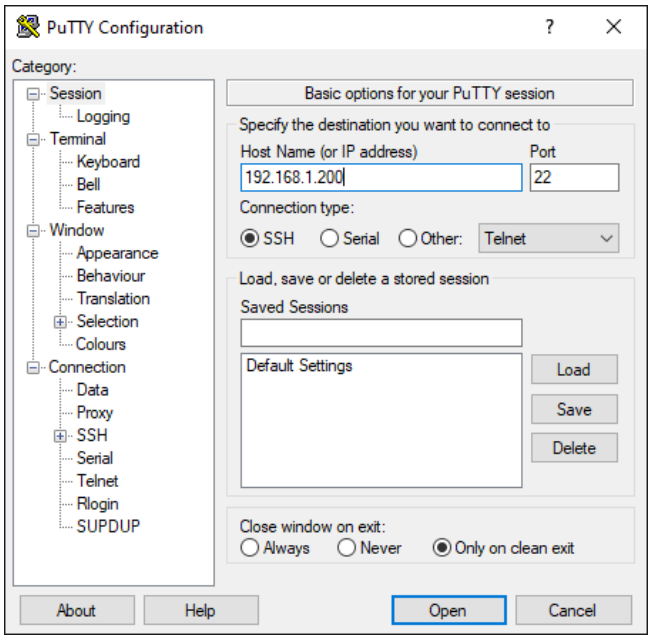
Conexión SSH

```
ruben@vega:~$  ssh usuario@192.168.1.200 -p 22
```

En el comando anterior podemos identificar:

- **ssh**: el cliente de consola
- **usuario**: el nombre del usuario con el que nos queremos conectar al servidor remoto.
- **@**: la arroba en inglés significa “at”, que indica “usuario en el servidor X”.
- **192.168.1.200**: La IP del servidor al que nos queremos conectar.
- **-p 22**: Estos dos parámetros van juntos, “-p” indica que vamos a indicar el puerto de conexión y “22” que nos queremos conectar a ese puerto. Debido a que 22 es el puerto por defecto, podríamos no poner estas opciones si sabemos que el servidor escucha en el puerto 22.

Si realizamos la conexión a través de un cliente de interfaz, como es putty, el aspecto será el siguiente, donde sólo podremos introducir la IP del servidor. Cuando se comience con la conexión nos pedirá los credenciales de acceso.



Si es la primera vez que nos conectamos a un servidor mediante SSH nos saldrá un mensaje como el siguiente:

Conexión SSH

```
ruben@vega:~$  ssh usuario@192.168.1.200 -p 22

The authenticity of host '192.168.1.200 (192.168.1.200)' can't be established.
ECDSA key fingerprint is SHA256:uK9MOl0gLDhTtCrlcafc1zVObVA/vnOMn6TWFsQb23o.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Este "key fingerprint" es un identificador que está relacionado con el fichero de "**clave pública**" del servidor. Es como el DNI del servidor. La primera vez se nos guarda ese *fingerprint*, y en caso de que en una próxima conexión varíe, nos avisará. No suele ser habitual que este identificador cambie.

3.3.3. Conexión mediante certificados de clave pública/clave privada

Existe una alternativa a la hora de realizar una conexión SSH para que no nos pida la contraseña del usuario, y es **hacer uso de los certificados de clave pública y clave privada**. Este concepto de "clave pública y clave privada" viene de la **criptografía asimétrica**.

Este sistema de criptografía asimétrica hace uso de dos claves que están asociadas entre sí:

- **Clave privada**: Es la base del sistema criptográfico, y como su nombre indica, se debe de mantener en privado. **Nunca se debe de compartir**, ya que entonces se podrían hacer pasar por nosotros.
- **Clave pública**: Asociada a la clave privada, la clave pública puede ser compartida y enviada a otros ordenadores para poder realizar la conexión.

Para generar el par de claves se realiza con el siguiente comando:

Crear par de claves pública/privada

```
ruben@vega:~$  ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ruben/.ssh/id_rsa):

Enter passphrase (empty for no passphrase):
Enter same passphrase again:

Your identification has been saved in /home/ruben/.ssh/id_rsa
Your public key has been saved in /home/ruben/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:SPqPOYBmPb8PCFhcZgqcWZPZzaL5RNfMeKmHqebvC7E ruben@vega
The key's randomart image is:
+---[RSA 3072]----+
|o +oB o = .      |
| * B.+ = *       |
```

```
|  + + + =        |
| o o + = .       |
|. .o+.o S        |
|  +.+*o          |
| o  +Eo          |
|     .+=         |
|      *B+        |
+----[SHA256]-----+
```

El comando muestra los siguientes pasos:

1. Creación de la pareja de claves pública/privada haciendo uso del sistema criptográfico **RSA**.
2. Lugar donde se va a guardar la clave privada. Por defecto en `\~/.ssh/id\_rsa`.
3. Contraseña para securizar la clave privada. De esta manera, para poder usarla habrá que introducir dicha contraseña. Dado que nosotros queremos evitar introducir contraseñas, lo dejaremos en blanco.
4. Lugar donde se va a guardar la clave pública. Por defecto en `\~/.ssh/id\_rsa.pub`

Una vez tenemos nuestro par de claves, podemos copiar la clave pública al usuario del servidor que nos interese mediante el siguiente comando:

Crear par de claves pública/privada

```
ruben@vega:~$  ssh-copy-id user@servidor_remoto
```

Para ello es imprescindible conocer previamente la contraseña del usuario en el servidor. El comando `ssh-copy-id` realizará una conexión SSH y copiará el contenido de la **clave pública**, `\~/.ssh/id\_rsa.pub`, dentro del fichero `\~/.ssh/authorized\_keys` del usuario en el servidor remoto. Este paso se puede realizar a mano (con un editor de texto).

¡Cuidado!

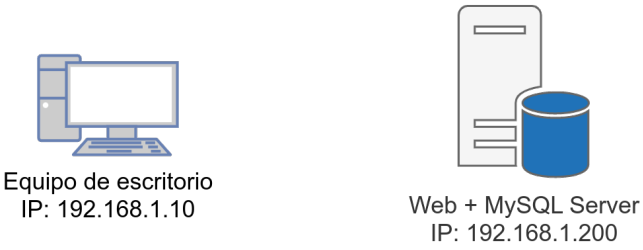
Windows no tiene el comando "ssh-copy-id", por lo que deberemos hacer el paso a mano, tal como se ha explicado.

Al realizar la siguiente conexión, ya no necesitaremos introducir la contraseña del usuario, ya que el sistema remoto podrá autenticarnos a través del sistema clave pública/privada.

3.3.4. Crear túneles SSH

Una de las funcionalidades extra de SSH es la posibilidad de crear "túneles" para securizar protocolos no seguros, o poder acceder a servicios que sólo escuchan en *localhost*.

Pongamos como ejemplo el siguiente escenario:



Tenemos un servidor web con Apache y MySQL al que queremos acceder. Por seguridad MySQL **sólo está escuchando en localhost (127.0.0.1)**, por lo que el acceso al servicio MySQL no es posible. Para administrarlo nos tenemos que conectar al servidor, y realizarlo de manera local.

En este punto es donde entra en juego la creación de un túnel seguro al servidor, para poder acceder al servicio remoto. **Para ello es imprescindible poder realizar una conexión SSH** (ya sea mediante usuario o clave pública/privada).

Para crear un túnel, desde el equipo de escritorio, lanzaremos el siguiente comando:

Crear par de claves pública/privada

```
ruben@vega:~$  ssh usuario@192.168.1.200 -L 6306:127.0.0.1:3306 -N
```

Al ejecutar este comando, habremos creado un túnel que enlaza el puerto remoto 3306 (que sólo se escucha en el "127.0.0.1" del servidor), con el puerto local 6306 del equipo de escritorio. A continuación la explicación del comando y sus parámetros:

- "**ssh usuario@192.168.1.200**": es como una conexión SSH normal. Lo que estamos indicando es que queremos conectarnos con "usuario" al servidor remoto 192.168.1.200 a través de SSH.

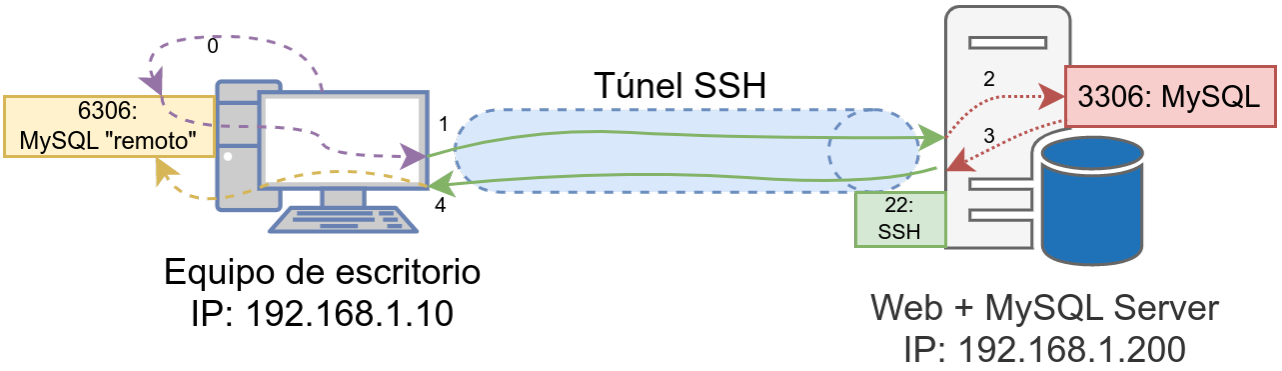
- **-L 6306:127.0.0.1:3306**: Especifica que el puerto local especificado se va redirigir al puerto e IP remota. Para entender esto hay que separar dos partes de los parámetros:

  - **6306**: Especifica la IP y el puerto local. En este caso, antes del puerto no hemos especificado IP, por lo que se creará un puerto 6306 que sólo se pone a la escucha en ***localhost*** en el equipo de escritorio

  - **127.0.0.1:3306**: Esta es la dirección y puerto remoto al que nos queremos conectar. En este caso, es el puerto 3306 que está escuchando en la IP 127.0.0.1 del servidor.

- **-N**: Sirve para que no ejecute ningún comando en el servidor remoto, y por tanto no nos abrirá conexión de terminal.

A nivel visual, y para entender de mejor manera lo realizado, sirva la siguiente imagen y los pasos que se pueden dar en un escenario real:



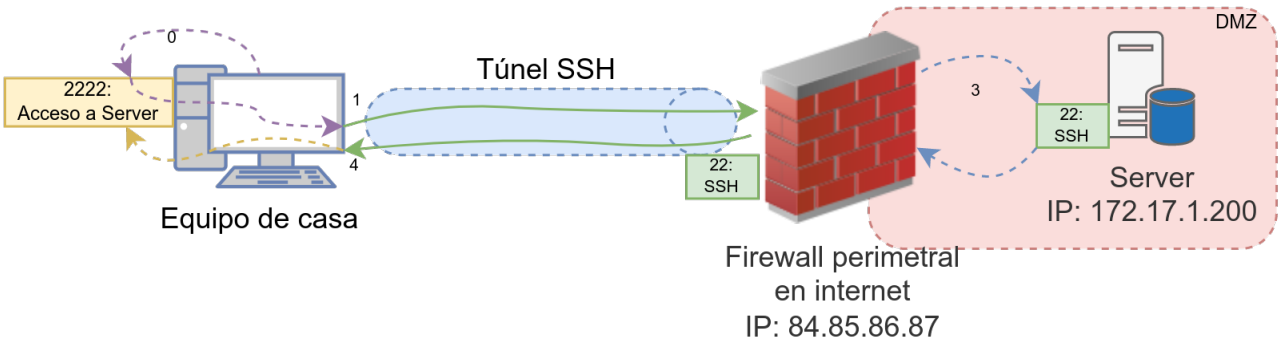
Con una aplicación en el equipo de escritorio queremos conectarnos al servidor MySQL que sólo escucha en el servidor remoto. Ejecutamos el túnel visto anteriormente, y los pasos que podremos realizar son los siguientes:

0. Mediante una aplicación nos podemos conectar al puerto local 6306, que ha sido creado mediante el comando anterior. Este puerto local está redirigido al puerto del servidor remoto. Por lo tanto la conexión se securiza a través del túnel
1. Como el túnel está establecido, la conexión viaja de manera segura a través de él.
2. Al llegar al servidor remoto, sabe que la conexión debe ir al puerto 3306 de la IP 127.0.0.1, que es lo establecido en el comando.
3. La conexión vuelve al túnel.
4. Viaja por el túnel hasta llegar a la comunicación que se había establecido previamente.

De esta manera, hemos podido realizar una conexión a un servicio remoto a través de SSH y completamente seguro.

3.3.4.1. Acceder a un equipo saltando a través de otro.

Este es un caso especial de túnel, similar a lo explicado previamente. En lugar de querer realizar una conexión a un servicio del equipo al que nos conectamos, en este caso lo usaremos de salto para acceder a otro servidor.



En este caso estando en casa nos queremos conectar a un equipo de la oficina. No tenemos VPN, y no hay redirección de puertos directa, pero podemos acceder al firewall perimetral. Por lo tanto, lo podemos utilizar para saltar al servidor que nos interesa.

En este caso, el comando a ejecutar sería:

Crear par de claves pública/privada

```
ruben@vega:~$  ssh usuario_firewall@84.85.86.87 -L 2222:172.17.1.200:22 -N
```

De esta manera, ahora desde nuestro equipo podremos realizar una conexión SSH al puerto 2222 que realmente será una redirección que viaja a través del túnel hasta el firewall, y que a su vez redirige al puerto 22 del servidor 172.17.1.200.

¡Atención!

El servicio remoto al que nos queremos conectar debe escuchar en la IP del equipo al que nos queremos conectar. El equipo que usamos para saltar debe poder conectarse a él.

4. Gestión de copias de seguridad (backups)

4.1. Introducción

Una copia de seguridad, o ***backup***, **es una copia de los datos originales** que se realiza con el fin de disponer de un medio **para recuperarlos en caso de pérdida**. Las copias de seguridad son útiles en caso de que los datos originales se hayan perdido por un fallo humano, fallo de sistema (rotura de disco duro), modificado por un virus, evento natural (incendio del edificio), un ataque (borrado deliberado de los datos), etc…

En el proceso de copia de seguridad también **se tiene que tener en cuenta el método de restauración de los datos**, que es el momento en el que necesitaríamos realizar la restauración de los datos del *backup* para dejarlos en la ubicación original.

A la hora de pensar un sistema de copias de seguridad, tendremos que tener en cuenta los siguientes puntos:

- Importancia de los datos
- Espacio ocupado por la copia de seguridad
- Ubicación de la copia de seguridad
- Plan de recuperación ante desastre
- Punto máximo que podemos recuperar
- Tiempo estimado de recuperación de los datos

Por otro lado, también tendremos que tener en cuenta el método que vamos a utilizar para realizar la copia de seguridad, y cómo va a ser el proceso. Este proceso dependerá del servicio/sistema al que vayamos a realizar el backup.

Y por último, la estrategia a utilizar:

- Estrategia multi-backup completos
- Incrementales y/o diferenciales
- Estrategia personalizada
- La que el programa que vayamos a utilizar nos permita
- …

Antes de realizar la copia de seguridad de los datos tendremos que tener muy en cuenta distintos apartados, que deberán ser analizados en detalle y puede que modificados en el futuro.

4.2. A tener en cuenta

Durante la **planificación** de la creación de las copias de seguridad hay ciertos aspectos que tenemos que tener en cuenta para que la gestión de copias de seguridad se realice de manera satisfactoria.

Esta planificación debe realizarse conociendo los datos que maneja la empresa, la importancia de los mismos, el volumen que ocupan, la cantidad de cambios que reciben, ...

Información

Una buena planificación para la gestión de copias de seguridad de una empresa puede involucrar a varias personas de distintos ámbitos internos.

4.2.1. Conocer los datos y la importancia de los mismos

Para saber sobre qué datos tenemos que realizar las copias de seguridad, debemos conocer los datos que manejamos en la empresa, la importancia que tienen y su ubicación. **Ejemplo**: no es lo mismo la importancia de los datos gestionados en la base de datos de clientes, o el directorio "Descargas" el PC de un usuario.

También tendremos que conocer el tamaño actual de esos datos, y saber cuánto volumen total alcanzan. Cuanto mayor tamaño tengan, también mayor será el espacio ocupado por las copias de seguridad.

Otro aspecto a tener en cuenta es saber las modificaciones que sufren esos datos, y estimar el volumen de esos cambios. **Ejemplo**: no es lo mismo que una base de datos reciba 10 clientes nuevos al día o que reciba 15.000 ventas a la hora en nuestra tienda online.

Información

Debemos conocer los datos que existen en nuestra empresa, la ubicación, la importancia y el volumen que ocupan para así poder realizar un buen plan de gestión de copias de seguridad.

4.2.2. Espacio ocupado por la copia de seguridad

Estimar el espacio que nos va a ocupar una copia de seguridad puede parecer sencillo, pero no siempre es así, y es por eso que conocer los datos, tal como hemos dicho antes, es de vital importancia.

Tendremos que tener en cuenta cuánto ocupan nuestros datos originales, cuál va a ser la estrategia de backup a utilizar, estimar el incremento de espacio que puede haber en los datos originales…

Por lo tanto, siempre deberíamos estimar por alto el espacio ocupado, y siempre pudiendo realizar la expansión de ese espacio "en caliente", para que las copias de seguridad funcionen cada día.

¡Cuidado!

No realizar copias de seguridad debido a falta de espacio debería ser considerado un fallo de primer orden y que debe ser subsanado lo antes posible.

4.2.3. Punto mínimo en el tiempo que queremos recuperar

Es posible tener copias de seguridad que estén completamente actualizadas en cuanto se realizan cambios en los datos originales, pero la puesta a punto de estos sistemas puede no estar a nuestro alcance (por no

conocer cómo se realizan o por un coste elevado).

En ciertos casos, algunas empresas pueden aceptar la pérdida de ciertos datos, ya que el gestionar que las copias se realicen “en tiempo real” es más caro que el repetir el proceso de crear esos datos de nuevo.

Ejemplo: Una empresa puede no necesitar hacer backup de la base de datos de cuándo fichan los empleados en tiempo real, porque si se pierden los datos de un día es asumible (el backup se hace cada noche). En cambio, la base de datos de marketing tiene que realizarse en tiempo real.

4.2.4. Punto máximo en el tiempo que queremos recuperar

Otro aspecto importante a tener en cuenta es **pensar cuánto tiempo para atrás queremos poder llegar para poder realizar la restauración de los datos**. No será lo mismo querer realizar una restauración de los datos de hace una hora, un día, de hace una semana o de hace un año, ya que esta decisión influirá en toda estrategia de copia de seguridad.

Es por eso, que a los datos hay que darle la importancia que se merece, y es bastante probable que distintos datos tengan distintos puntos de restauración máximo.

Ejemplo: No es lo mismo los datos de facturación de una empresa, que la ley nos marca que debemos guardar dichos datos (4 años), los datos de una página web que apenas cambia (igual nos sirve una copia a la semana) o la carpeta personal de un usuario en su equipo (dado que en su equipo no debería guardar nada importante de la empresa, se podría asumir una pérdida de datos).

4.2.5. Ubicación de la copia de seguridad

Es muy importante también tener en cuenta cuál va a ser la ubicación de la copia de seguridad y las consecuencias que eso puede conllevar. Vamos a poner varios ejemplos:

- Si decidimos hacer la copia de seguridad en el mismo equipo informático, un problema de electricidad podría suponer la pérdida de los datos originales y los backups.
- Si decidimos realizar la copia de seguridad en la misma sala de ordenadores donde se sitúan los datos originales, si esa sala se ve involucrada por un factor que destruya todo (un incendio, un robo, …), perderíamos todos los datos.
- Si decidimos realizar la copia en la misma oficina, edificio, puede pasar lo mismo que el punto anterior
- Si decidimos realizar la copia en la “nube”, el tiempo para realizar la copia de seguridad y el tiempo de restauración de los datos puede incrementarse
- Si decidimos realizar la copia en otra oficina, dependemos de la conexión, los datos deberían ir cifrados, tiempo de creación de la copia de seguridad y restauración…

Como se puede ver, cada ejemplo puede tener unos pros y unos contras.

¡Cuidado!

La copia de seguridad nunca debería estar en el mismo equipo informático, y mucho menos en el mismo disco duro físico, que los datos originales.

4.2.6. Tiempo estimado de la copia

Teniendo en cuenta lo comentado previamente, el tiempo estimado de la copia de seguridad es muy importante, y es posible que varíe en el tiempo. No es lo mismo realizar la copia de seguridad de unos pocos megabytes o de varios gigabytes.

Es por eso por lo que tendremos que tener en cuenta el tiempo que tarda en realizarse nuestras copias de seguridad para así asegurar que se realizan de manera correcta y no se solapan en el tiempo.

4.2.7. Plan de recuperación ante desastre

Tan importante es tener una copia de seguridad como un plan ante un posible desastre. De nada sirve tener una copia de seguridad si restaurarla nos va a llevar semanas por no saber el estado de las mismas, o si su restauración es compleja.

Por lo tanto, **es muy importante disponer de un plan de recuperación ante desastres que esté actualizado**, que contemple si ha habido modificaciones en la gestión de copias de seguridad, que tenga en cuenta los pasos a realizar…

Este plan de recuperación **debería ser puesto en práctica cada cierto tiempo para asegurar que sigue estando vigente**, y si no, realizar las modificaciones oportunas.

¡Atención!

Es muy importante disponer de un plan de recuperación ante desastres que esté actualizado y probar que siga vigente cada cierto tiempo.

El plan de recuperación ante desastre debería contener:

- Los distintos métodos y estrategias de copias de seguridad que existen en la empresa
- Ubicaciones donde se realizar las copias de seguridad
- Importancia de los datos de los que se realizar la copia de seguridad
- Cómo realizar la restauración de cada sistemas de los que se ha realizado la copia de seguridad
- Orden para realizar las restauraciones y dada la importancia de los datos, la prioridad de las restauraciones
- ...

Información

El plan de recuperación es algo que toda compañía debe tener a la hora de crear sus sistema de copias de seguridad, ya que deben de ir de la mano.

4.2.8. Tiempo estimado de recuperación de los datos

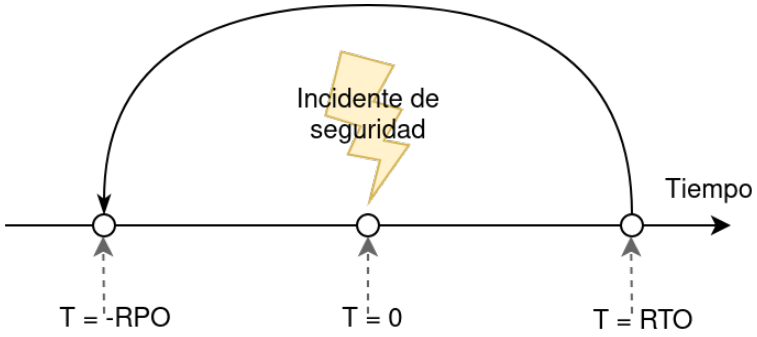
El último punto, pero no por ello menos importante, es el tiempo estimado de recuperación de los datos.

¡Cuidado!

De poco sirve tener copias de seguridad y un plan de recuperación, si luego es posible que estemos dos semanas para recuperar los datos.

Es por eso que tenemos que conocer el tiempo estimado de recuperación de los datos para poder gestionar los tiempos con los empleados, los clientes…

Este tiempo estimado, al igual que el tamaño de las copias, puede variar en el tiempo, por lo que **debemos asegurar cada cierto tiempo el realizar simulacros para asegurar qué tiempos se manejan**.



Teniendo en cuenta el gráfico anterior:

RTO: es el objetivo de tiempo de recuperación (*Recovery Time Objective*), y es el tiempo que se tarda en restablecer el servicio, o al menos los mínimos acordados. Cuanto menor sea el tiempo de recuperación, mejor.

RPO: es el objetivo de punto de recuperación (*Recovery Point Objective*), y es el último instante de tiempo previo al incidente al que los sistemas son capaces de regresar. Habitualmente suele ser marcado por la frecuencia con que se realicen las copias de seguridad. De nuevo, cuanto más cercano a T=0 mejor.

4.3. Métodos de Backup

Existen distintas estrategias a la hora de crear un backup, por lo que tendremos que analizar los datos que queremos salvaguardar y decidir la mejor estrategia.

4.3.1. Copiar los ficheros

Sería la metodología más sencilla. Simplemente haremos una copia de los datos importantes en otra ubicación/soporte. Este sistema hace que cada X tiempo, se guarden los ficheros y obtendremos una copia

completa de los datos.

El problema es que **no tendremos la posibilidad de recuperar un fichero en un estado anterior a la última copia**. Es decir, si se realiza la copia de seguridad a las 6:00 de la mañana, no podremos obtener el estado del fichero previo a ese. Si a las 11:00 el usuario corrompe el fichero, podríamos recuperar al estado de las 6:00, pero no al estado de ayer.

4.3.2. Sistema incremental a nivel de bloque

Se basa en copiar sólo los bloques físicos que han sido modificados. Tendremos que tener guardado el fichero original por un lado, y a partir de ahí sólo guardaremos las modificaciones de manera incremental.

Para recuperar el fichero, primero tendremos que elegir restaurar el fichero original completo, y posteriormente aplicar los bloques incrementales hasta llegar al punto de restauración que nos interesa.

4.3.3. Sistema incremental o diferencial binaria

Similar al anterior, pero esta vez en lugar de hacer uso de bloques (1K, 4K, 8K…), la parte incremental o diferencial sería a nivel binario.

4.3.4. Versionado de ficheros

El versionado de ficheros se puede realizar de distintas maneras:

- **Utilizando el sistema de ficheros**: Existen sistemas de ficheros que cuando uno es modificado, automáticamente crea una versión nueva del mismo.
- **Versionado manual**: cuando el usuario quiere crear una nueva versión del fichero, debe ejecutar a mano una acción para guardar la nueva versión del mismo.

4.4. Estrategias de backup

Existen muchas estrategias a la hora de realizar una copia de seguridad, quizá tantas como entornos existen, ya que se pueden utilizar estrategias generales o crear una para cada caso concreto. Vamos a explicar las más habituales.

4.4.1. Multi-backup completos

Este método sería el más sencillo, pero también el que más espacio nos ocuparía, y posiblemente el que más tiempo puede tardar, dependiendo de los datos a copiar.

En este método, podríamos hacer una copia completa de los datos que queremos guardar en tantas ubicaciones como queramos. Supongamos que queremos poder llegar a restaurar hasta 3 días de un fichero corrupto, eso supondría que tendríamos que tener 3 ubicaciones distintas donde cada día haríamos un backup completo de los datos. Por ejemplo:

- Ubicación **1**: el **lunes** haríamos el primer backup completo
- Ubicación **2**: el **martes** haríamos el segundo backup completo

- Ubicación **3**: el **miércoles** haríamos el tercer backup completo
- Ubicación **1**: el **jueves** haríamos el backup completo (machando los datos previos que había del lunes)
- …

Como se puede ver, es una estrategia cíclica, sencilla, y que en ciertos casos puede ser más que suficiente. Pero como se ha comentado previamente, cada día se realizará un backup completo, por lo que habrá que tener en cuenta el tiempo que tarda, el espacio que ocupa, ...

4.4.2. Incrementales y/o diferenciales

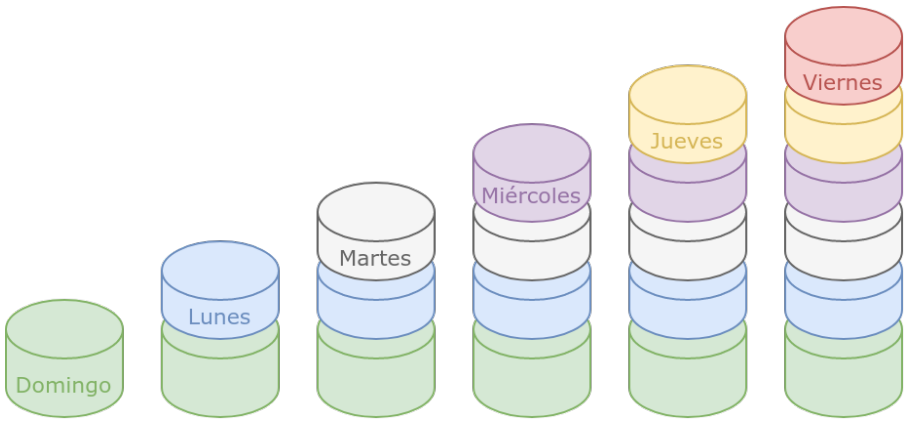
Normalmente es el método más habitual y todos los programas suelen tener estos métodos a la hora de realizar copias de seguridad.

El hacer uso de copias incrementales y/o diferenciales nos permitirá poder restaurar los datos hasta un punto concreto. Por eso será muy importante tener en cuenta cuánto tiempo queremos poder llegar a restaurar, porque no es lo mismo a la hora de estimar espacio ocupado, si queremos restaurar los datos de una semana, de hace un mes o de hace un año.

Por lo tanto, a la hora de utilizar esta metodología, como ya se ha comentado, que es la más habitual, la estimación de espacio debe de ser holgada, y pensar a futuro, en cuánto puede llegar a aumentar el espacio ocupado.

4.4.2.1. Incrementales

Supongamos que realizamos una copia completa de los datos el domingo, esta será la base para las siguientes copias incrementales de los datos:

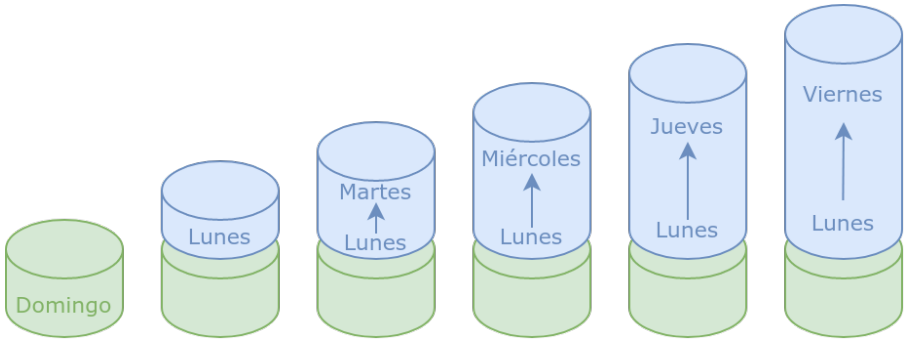


El lunes se hará una copia guardando sólo los datos que han cambiado respecto a la copia del domingo. El martes, se hará una copia de los datos que han sido modificados desde la copia del lunes… Y así sucesivamente.

Este método es rápido, fácil de realizar y sabemos claramente cuándo se han modificado los datos, por lo que si tenemos cada incremental separado en distintas carpetas, la restauración sería también sencilla.

4.4.2.2. Diferencial

Similar al incremental, pero cada día se guarda todos los datos que han sido modificados desde el último backup completo:



Esto hace que cada backup diferencial pueda ocupar más que un incremental, pero cada día obtendremos todos los cambios desde el último backup completo.

4.4.2.3. Incremental y diferencial

También suele ser habitual realizar la mezcla de ambas estrategias:

- El primer domingo de cada mes realizar la copia completa.
- Hacer uso del método incremental de lunes a sábado
- Los domingos realizar un diferencial de toda la semana.

4.4.3. Estrategia personalizada

Como se ha comentado previamente, pueden existir tantas estrategias como entornos existan, por lo que dependiendo de nuestros datos, nuestro entorno, nuestro modelo de negocio, tendremos que crearnos una estrategia propia.

Este método puede ser el más complejo, ya que quizá tengamos que realizarlo a mano, pero con ello nos aseguraremos de darle la importancia que se merece a cada parte de los datos que tengamos que realizar un backup.

4.4.4. La que el programa nos marque

Si hacemos uso de un programa de gestión de backups, al final estaremos limitados a lo que el programa nos permita. Como se ha comentado, la estrategia incremental/diferencial suele ser la más ampliamente utilizada, por lo que casi todos los programas deberían tener esta opción.

Dependiendo del programa, nos permitirá hacer mezclas de estrategias, realizar copias a sistemas remotos (la nube, por FTP, SSH…), varios cada día… Por lo que quedaría en nuestra mano analizar distintos programas y ver cuál es el que mejor se adecúa a nuestras necesidades.

4.5. Regla 3-2-1

Con todo lo dicho anteriormente, una de las estrategias más habituales, de manera generalizada y simplificada, es la conocida como la "**Regla 3-2-1**".

Esta regla se resume en:

- **3 copias completas** de los datos.
- En **2 tipos de almacenamiento distintos**
- **1 de las copias debe estar fuera**

Imaginemos una base de datos de una tienda online que está situada en un servidor en un RACK junto con otros servidores nuestros en un proveedor contratado. En esta base de datos tenemos **los datos originales**.

Haremos uso de otro de los servidores del RACK en el que realizaremos una copia de seguridad completa. Esta será la **segunda copia completa**, que está en otro servidor y que podremos acceder de manera rápida a ella.

La **tercera copia** de los datos la tendremos en nuestra oficina, ya que realizaremos una copia remota de los datos. Esta tercera copia está situada fuera del proveedor contratado, por lo que si en ese CPD hubiese un incendio, tendríamos una copia en local.

4.6. Realización de simulacros de recuperación de los datos

De nada sirve lo comentado hasta ahora, si no se realizan simulacros de desastre y de recuperación de los datos para asegurar que todo funciona de manera correcta. Desgraciadamente, **también puede haber fallos en los sistemas de backups**, por lo que aunque creamos que se están realizando correctamente, quizá el día que vayamos a necesitarlos nos damos cuenta que no es así, por lo que no podríamos realizar la recuperación de datos.

¡Cuidado!

Deberíamos crear simulacros de desastre para asegurar que todos los eslabones de la cadena de nuestro sistema de copias de seguridad es funciona de manera correcta.

Estos simulacros nos deberían ayudar a asegurar que:

- Las copias de seguridad funcionan de manera correcta
- Podemos realizar restauraciones de los datos a fecha que nos interesa
- El plan de recuperación ante desastre está actualizado y es funcional
- El tiempo estimado de recuperación es el adecuado una vez restaurados los datos, nuestro sistema es funcional y podemos seguir con la actividad comercial

¡Atención!

Si alguno de los puntos anteriores falla y no es correcto, deberemos repasar todo el plan estratégico de copias de seguridad para solventar los fallos, y así poder volver a realizar un nuevo simulacro para confirmar que esta vez es correcto.

5. Sistemas de Monitorización

El sistema de monitorización se encarga de recopilar información acerca del estado de los servidores de una infraestructura. Entre las métricas y datos que debe recopilar se encuentran:

- **Estado del servidor**: cantidad de RAM utilizada, estado de los discos duros, carga del servidor, sistema de ficheros, …
- Estado de los **servicios que tiene el servidor**: servidor web, número de procesos que tiene, bases de datos, conexiones existentes, cola de correos electrónicos para enviar si es un servidor de correo, … Dependiendo del servicio habrá que realizar unas comprobaciones u otras.
- **Infraestructura en la que se encuentran**: estado de la red, conexión a otros servidores, …
- **Estado del clúster**: en caso de que el servidor pertenezca a un clúster, hay que comprobar que el clúster se encuentra en perfecto estado.
- **Dependencias externas**: un servidor puede depender a su vez de otros, o de servicios externos, que deben de estar funcionando de manera correcta.

Normalmente en la monitorización actúa un **agente** (o servicio) instalado en el equipo monitorizado que obtiene la información requerida que se envía a un servidor central que recopila la información de toda la infraestructura monitorizada.

Esta información suele ser almacenada durante un periodo de tiempo determinado (un año, por ejemplo) para poder ser usada y comparar la situación de los servidores a lo largo del tiempo. Gracias a esta comparación temporal **se puede llegar a predecir el estado del servidor** a unos días/semanas vista y evitar problemas antes de que sucedan (no tener espacio en discos duros, mal funcionamiento de servicios por falta de RAM, …).

Información

La monitorización de servicios y equipos dentro de una infraestructura debe considerarse parte del proyecto, ya que es una parte muy importante de cara al mantenimiento del mismo.

5.1. Monitorización de servidores

Es habitual que los sistemas de monitorización funcionen en base a plantillas, que posteriormente se pueden asociar a los servidores monitorizados. Estas plantillas contendrán los servicios que deben ser monitorizados en cada tipo de servidor, ya que no es lo mismo monitorizar un servidor web o un servidor con un SGBD.

Para monitorizar un servidor lo habitual suele ser realizar las siguientes operaciones:

- **Comprobar conectividad con el servidor**: suele ser habitual que los servidores estén fuera de nuestra red (en un proveedor de Internet, en un cliente, en otra oficina…), por lo que es necesario

que exista conectividad de alguna manera para poder realizar la monitorización. En caso de no estar en nuestra red, el uso de una VPN es lo más utilizado.

- **Instalar un agente en el propio servidor**: Será el encargado de recopilar la información necesaria para mandarla al servidor central. Dependiendo del sistema de monitorización utilizado, necesitaremos un tipo de agente u otro. Algunos se encargan de realizar todas las comprobaciones y otros llaman a otros programas para realizar las comprobaciones y después mandar el resultado al servidor central.
- **Dar de alta el servidor en el sistema centralizado**: Tal como se ha dicho previamente, lo habitual es contar con un sistema centralizado en el que se tendrán todos los servidores y el estado de las comprobaciones realizadas. De ser así, habrá que darlo de alta, y para ello se necesitará:
  - **Nombre del servidor**: Un nombre que a simple vista identifique el servidor. Suele ser habitual poner el nombre del cliente también, y/o el tipo de servicio que preste.
  - **IP del servidor**: Para poder realizar la conexión al servidor.
  - **Plantillas asociadas**: En caso de utilizar un sistema que utilice plantillas, al dar de alta el servidor se le aplicarán las plantillas necesarias para que realicen todos los checks oportunos. Por ejemplo: plantilla de Servidor Linux + plantilla de Servidor web + Plantilla de MySQL.
  - **Puerto de conexión**: Los agentes de monitorización suelen contar con un puerto que queda a la escucha. Si hemos cambiado el puerto, habrá que indicarlo a la hora de dar de alta.
  - **Otras opciones**: Dependiendo del sistema de monitorización se podrán añadir muchas más opciones, como por ejemplo:
    - **Servidores de los que se depende**: Imaginemos que el servidor monitorizado depende a su vez de un router que también está monitorizado. Si el router cae, no llegaríamos al servidor, por lo que realmente es una caída por dependencia, aunque el servidor puede estar funcionando de manera correcta. Esto nos puede permitir crear “árboles de dependencias” de servidores.
    - **Periodos de monitorización**: Lo habitual es que un servidor esté monitorizado 24x7, pero quizá nos interese realizar cambios y que sólo se monitorice en unas horas determinadas (quizá el resto del tiempo está apagado).
    - …

5.2. Funcionamiento de la monitorización

Para conocer cómo funciona un sistema de monitorización lo mejor es que tomemos como ejemplo un tipo de servicio que queremos monitorizar. Como ejemplo se puede tomar las comprobaciones que queremos realizar a un SGBD (Sistema Gestor de Bases de Datos).

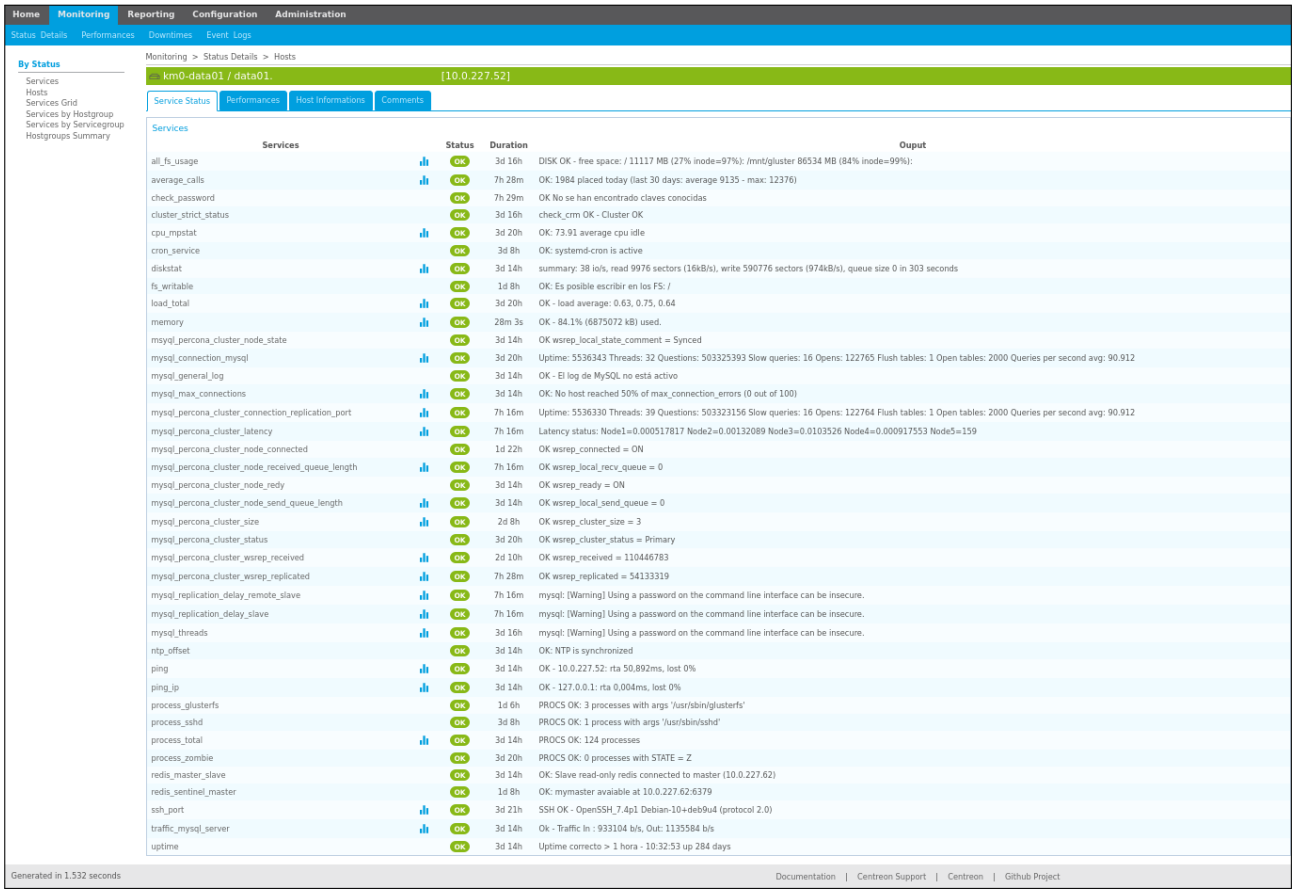
No será lo mismo realizar la monitorización de un servidor MySQL o de un Oracle, pero las comprobaciones

que queremos realizar en ellos deberían ser similares. Vamos a querer realizar la monitorización de las mismas comprobaciones: estado de las tablas en memoria, número de hilos en ejecución, número de *slow_queries*, …; pero los scripts ejecutados serán distintos.

Información

La monitorización dependerá del propio servicio que vayamos a monitorizar.

A continuación se puede ver el estado de un servidor monitorizado a través del sistema de monitorización Centreon:



Servidor monitorizado en Centreon

En la imagen anterior se puede comprobar un número de ***checks***, o comprobaciones, que se están realizando sobre un servidor concreto. Cada fila es una comprobación y contienen:

- **Nombre del check/servicio**: Un nombre para identificar qué es lo que se está comprobando con el check.
- **Icono para mostrar gráficas**: Algunos checks recibirán información que puede ser graficada para así poder observar patrones en el comportamiento del servidor. Por ejemplo: cantidad de RAM ocupada, número de procesos en el sistema, número de conexiones a un servidor, …
- **Estado del check**: Normalmente, tras realizar la comprobación, el check termina con uno de los

siguientes resultados:

- **OK**: El resultado obtenido es el correcto.
- **Warning**: El resultado obtenido está entre los márgenes de peligro. Es posible que de seguir así pase al estado siguiente:
- **Crítico**: El servicio devuelve un estado que es considerado crítico, lo que puede hacer que llegue a mal funcionamiento del mismo, o incluso que el servidor comience a dejar de funcionar (imaginemos que el servidor está con el 90 % de la RAM ocupada o de disco duro ocupado).
- **Indeterminado**: Por alguna razón, el *check* no se ha realizado, o el valor devuelto es indeterminado o no se puede saber en qué otro estado situarlo.

- **Duración del estado**: Para conocer cuánto tiempo lleva en el estado la comprobación obtenida. Lo ideal es que nunca haya estados que no sean OK y por lo tanto la duración de los mismos sea lo más alta posible.
- **Valor devuelto por la monitorización**: El valor real devuelto por la comprobación realizada. En base a este resultado se puede realizar las gráficas mencionadas previamente.

Información

El estado del servicio dependerá del valor devuelto por la monitorización.

Este resultado se cotejará con los valores que hayamos puesto para que sea considerado OK, Warning o Critical. Es decir, **en algunos casos el estado del servidor depende de los valores devueltos y de la baremación que le hayamos otorgado**.

Pongamos como ejemplo la monitorización de un SGBD:

- **El servicio del SGBD está funcionando**: Ahí no hay baremación posible. Si el servicio no está arrancado, es lógico pensar que el estado es crítico y que por tanto hay que ver qué ha ocurrido.
- **Número de conexiones en el SGBD**: El resultado devuelto será un número entero (que podremos graficar para obtener patrones). En este caso, podemos decidir los rangos para que el resultado sea OK, Warning o Critical. Es decir, si el resultado obtenido está por debajo del umbral de Warning, el sistema considerará que el estado es OK. Si está en dicho rango, será Warning y si está en el rango de Critical, así lo indicará.

Esta baremación y **estos rangos** se suelen aplicar también en las plantillas de los servicios. Hay que entender que también **pueden ser modificados y personalizados para un servidor concreto**. No es lo mismo que un SGBD tenga 500 conexiones simultáneas si tiene 8Gb de RAM o si tiene 128Gb (en el primer servidor se puede considerar que es un estado crítico mientras que en el segundo es lo esperado).

Cuando un *check* termina siendo un Warning o un Critical **es habitual que haya un sistema de alarmas configurado**. Dependiendo del sistema utilizado, notificará a los administradores mediante e-mail,

mensajería instantánea, SMS, … para que realicen un análisis lo antes posible y solucionen el estado del servicio.

Información

Los sistemas de monitorización suelen contar con un sistema de alarmas para que nos avise de los servicios caídos.

5.2.1. Monitorización básica

Tal como se ha comentado, en los servidores se suele realizar una monitorización del estado del mismo que suele ser común para todos, por lo que lo habitual suele ser tener una plantilla genérica para todos los servidores con la que se monitorizará:

- Cantidad de RAM utilizada
- Cantidad de memoria virtual utilizada
- Carga de la CPU
- Espacio libre en las unidades de disco duro
- Estado del sistema RAID del servidor (en caso de tenerlo)
- Cantidad de usuarios conectados a la máquina
- Estado de puertos de conexión (SSH, por ejemplo)
- Latencia hasta llegar al servidor
- …

Es cierto que no será lo mismo monitorizar un sistema GNU/Linux o un sistema Windows (ya que puede variar alguno de las comprobaciones a realizar), pero el estado general que queremos conocer es el mismo. Por lo tanto, lo habitual es tener dos plantillas, una específica para servidores Windows y otra para GNU/Linux.

5.2.2. Monitorización de Servicios

Aparte de la monitorización básica comentada previamente, necesitaremos monitorizar el estado de los servicios que pueda tener el servidor propiamente dicho. Para ello, de nuevo, se crearía una plantilla específica para cada tipo de Servicio que podamos tener en nuestro servidor.

No será lo mismo monitorizar un servidor que tenga un servidor web, un servidor de base de datos, un proxy… O puede que el servidor cuente con todos esos servicios.

Es por eso que a la hora de realizar la monitorización de un servidor **es muy importante conocer qué funciones desempeña cada servidor en la infraestructura a la que pertenece** y analizar los servicios que tiene arrancados para posteriormente ser monitorizados.

Información

Es muy importante conocer qué funciones desempeña cada servidor en la infraestructura a la que pertenece.

5.3. Tipos de monitorización

Existen varias maneras de realizar la monitorización de un servidor, y dependerá del gestor de monitorización que usemos (en caso de usar uno).

Es habitual que cuando nos referimos a sistemas de monitorización lo dividamos en dos grandes familias:

- Monitorización Activa
- Monitorización Pasiva

Estas dos maneras de monitorización suelen ser excluyentes, aunque algunos sistemas de monitorización permiten ambas, por lo que nos puede interesar usar una u otra dependiendo de la situación.

5.3.1. Monitorización pasiva

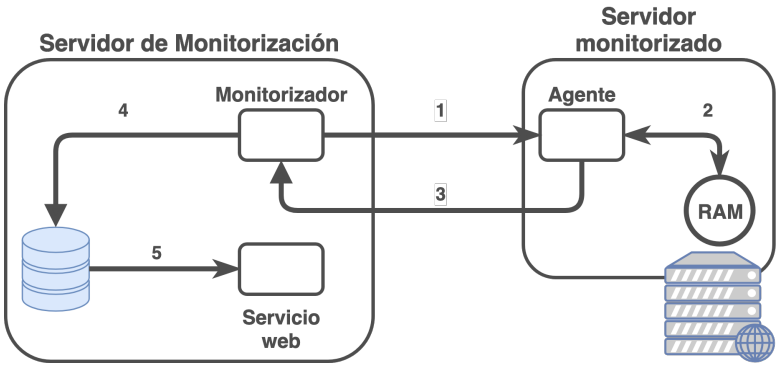
En la monitorización pasiva el servidor (u objeto monitorizado) es el encargado de mandar la información de manera periódica al servidor central. El agente instalado se ejecutará como una tarea programada cada cierto tiempo (habitualmente unos pocos minutos) e informará de la situación cambiante, de haberla, al servidor central.

Esta manera de monitorización es utilizada también cuando no hay un servidor central. En este caso, si la comprobación ha sido incorrecta, podría mandar un mail al administrador del servidor.

5.3.2. Monitorización activa

Suele ser la manera habitual de proceder de los sistemas que cuentan con un servidor centralizado de monitorización. El servidor de monitorización se encarga de preguntar al servidor, a través de la conexión con el agente, por la comprobación de alguno de los checks, y el agente devuelve la información.

A continuación se puede observar las etapas que existen en un sistema de monitorización activa utilizando un servidor de monitorización central:



Proceso de monitorización activa

Las etapas serían:

0. El sistema de monitorización tiene un scheduler (o planificador) que decide cuándo tiene que realizar cada comprobación (normalmente, cada pocos minutos).
1. El servicio encargado de monitorizar **establece conexión con el agente remoto** y le pide que compruebe un estado. En este ejemplo se ha optado por la RAM.
2. El agente en el servidor que se quiere monitorizar recibe la notificación y realiza una **comprobación local** (normalmente llamando a scripts locales) para obtener la cantidad de RAM ocupada, total y libre que tiene
3. El agente envía al sistema de monitorización el resultado obtenido en la ejecución de los scripts del paso anterior.
    1. El monitorizador al recibir el resultado, lo coteja con los rangos de baremación que tiene y decide si el check está en estado OK, Warning o Critical.
    2. Lo habitual es que si el resultado del servicio no es OK, se ejecute en el servidor de monitorización algún tipo de alarma (ya sea enviar un mail, sistema de mensajería, …) para notificar a los administradores.
4. El sistema de monitorización guarda en una base de datos los resultados obtenidos para así poder realizar posteriores análisis o comprobaciones temporales de los mismos.
5. Esos datos se suelen visualizar en una interfaz web, tal como hemos visto previamente.

Estos pasos son ejecutados de manera continuada en el servidor de monitorización para cada comprobación que se realiza en cada uno de todos los servidores que se monitorizan. Por lo tanto, se entiende que el propio servidor de monitorización también tiene que ser monitorizado ya que es de vital importancia que su estado sea óptimo.

5.3.3. Monitorización centralizada

Como ya se ha comentado, es el sistema habitual de monitorización. Las ventajas que podemos obtener al hacer uso de este sistema son muchas, pero se pueden destacar las siguientes:

- **Monitorización centralizada**: Aunque parezca obvio, el tener un único sistema en el que concentrar toda la información es muy útil y eficaz.
    - La alternativa sería tener una monitorización distinta en cada servidor.
- **Interfaz web**: Hoy en día suele ser habitual que los sistemas de monitorización tengan un servicio web en el que visualizar todos los datos obtenidos.
- **Sistema de plantillas**: De nuevo, es lo habitual, lo que hace que la gestión de monitorización de servidores sea más cómoda.
- **Gestión de usuarios**: Podremos tener usuarios que puedan ver unos servidores u otros, por lo que podemos tener equipos especializados en distintos grupos de monitorización y que sólo se enfoquen en ellos.

- Esto también es útil para dar acceso a los clientes a la monitorización de sus propios servidores.

5.3.4. Monitorización reactiva

La monitorización reactiva se puede definir como el sistema de monitorización que no sólo se encarga de comprobar y recibir el estado de los servidores, si no que también reacciona a los mismos para tratar de solucionar los problemas encontrados. Tras esta definición está la idea de que **existen ciertos fallos recurrentes que no siempre necesitan la intervención humana para solucionarse**, y que por tanto, se puede tratar de ejecutar antes de que sea considerado un problema real.

Como **ejemplo sencillo** se puede poner **el espacio libre en disco duro**. Imaginemos que se comprueba que apenas hay espacio en el disco duro de un servidor. En este caso, el sistema de monitorización recibirá que el servidor **está al 99.95 %** de espacio ocupado, y por tanto, en lugar de notificar a un humano indicando el estado crítico, **el sistema reacciona de manera automática tratando de liberar espacio**. Se habrá configurado previamente que en la reacción de este error trate de borrar ficheros temporales, vaciar papelera, limpiar ficheros de caché de ciertas rutas … Una vez hecho esto, se volverá a comprobar el estado del servidor. Si el espacio ocupado en disco duro ha bajado y está en modo OK no habrá que hacer nada más, y se habrá evitado que un administrador tenga que realizar dicha tarea. Si por el contrario el estado sigue siendo incorrecto, el sistema notificará el error para que se realice un análisis y se solucione el problema.

Como **ejemplo extremo** (que no suele ser habitual configurarlo así), imaginemos que **la RAM consumida por un SGBD es muy alta** y esté poniendo en peligro el estado del servidor, se podría configurar para que **el sistema reaccione reiniciando el SGBD para que libere la RAM** y vuelva a prestar servicio.

5.4. Gestores de monitorización

Hoy día existen muchos sistemas de monitorización, y dependiendo de nuestras necesidades deberemos optar por uno u otro. A continuación se expondrán varios ejemplos de gestores de monitorización basados en Software Libre, aunque la gran mayoría de ellos cuentan con un sistema dual. Es decir, se puede descargar y montarlo en tu propio servidor o puedes contratar a la empresa para que ellos tengan el servicio central:

- **Nagios**: Se puede considerar uno de los sistemas de monitorización más conocidos y del que se han basado otros. Generó mucha comunidad de administradores creando muchos scripts/plugins para hacerlos funcionar con él. Estos mismos scripts suelen ser utilizables en otros sistemas de monitorización.
- **Centreon**: Originalmente se creó como interfaz web para Nagios, pero poco a poco fue sustituyendo partes de Nagios hasta terminar siendo un sistema de monitorización completo. Existe la posibilidad de realizar la instalación por paquetes, descargar el sistema operativo en una ISO que te instala todo o incluso una máquina virtual con todo ya instalado y con configuración básica. (Demo).
- **PandoraFMS**: Sistema de monitorización creado por el español Sancho Lerena Urrea. Al igual que los anteriores, tiene sistema dual y la instalación se puede realizar por varios métodos.
- **Cacti**: Sistema más sencillo que los anteriores y habitualmente utilizado sólo en servidores sueltos,

es decir, no de manera centralizada.

- **Munin**: Igual que el anterior, ideal para monitorizar unos pocos servidores, ya que no se puede considerar un sistema centralizado como los primeros. Ver anexo de instalación de Munin.

Existen otros sistemas de monitorización basados “en la nube”, cuya funcionalidad es similar a lo expuesto previamente. Para hacer uso de estos sistemas nos descargamos un agente, lo instalamos y se encargará de mandar la información a los servidores de la plataforma contratada. Lógicamente, dependiendo del gasto realizado obtendremos más o menos servicios. Entre este tipo de servicios se pueden destacar:

- New Relic
- DataDog

6. Instalar sistema de monitorización Munin

A continuación se detalla cómo realizar la instalación y configuración básica de Munin como sistema de monitorización.

6.1. Introducción

Munin es un sistema de monitorización que se instala en cada servidor que queramos monitorizar. Es un sistema de monitorización sencillo que automáticamente monitoriza el propio servidor. Munin puede utilizarse para:

- Monitorizar el propio servidor
- Crear gráficas con los resultados de monitorización
- Usar plugins para monitorizar distintos servicios
- Realizar avisos sencillos por mail por situaciones anómalas

Munin necesita de un servidor web, como puede ser Apache, para poder visualizar las gráficas que crea con los datos obtenidos de la monitorización. Estas gráficas no son más que datos que se obtienen de la monitorización, que son almacenados en pequeñas bases de datos y que mediante un proceso CRON se generan las imágenes con dichos datos.

Aunque Munin puede ser suficiente para la monitorización de un servidor, no se suele utilizar en casos en los que se tenga más de 10-15 servidores, ya que hay que ir servidor a servidor para realizar la configuración, por lo que dificulta la administración.

Munin posee un sistema básico de monitorización centralizada, pero dista mucho de lo que se puede considerar un sistema de monitorización centralizada real. Tal como se ha dicho, para pocos servidores puede ser suficiente, pero no para infraestructuras grandes.

La ventaja que se obtiene al utilizar Munin es que es una instalación muy sencilla, tal como veremos a continuación, y que por tanto, para infraestructuras pequeñas puede ser útil.

6.2. Instalación

Para que Munin funcione se va a necesitar instalar el propio Munin y Apache como servidor web para mostrar la web en la que se visualizan los datos obtenidos de la monitorización.

Para realizar la instalación, ejecutaremos:

Instalar Munin y Apache2

```
root@vega:~# apt install munin apache2
```

Tras la ejecución de este comando, ya se habrá instalado los servicios necesarios para que Munin funcione y el servidor web Apache.

6.2.1. Configuración de Apache

La configuración para que Munin se pueda ver a través del servidor web Apache se encuentra en el fichero de configuración `/etc/munin/apache24.conf`. Normalmente esta configuración ya está aplicada en el Apache tras realizar la instalación, y se puede ver que existe un enlace simbólico, que si seguimos desde `/etc/apache2/conf-enabled/munin.conf` nos llevará al fichero mencionado previamente.

Esta configuración permite ver la web de Munin pero sólo si estamos conectados desde el propio servidor, a modo de seguridad sólo permite conexiones desde "localhost". Esto no suele ser lo habitual en servidores, por lo que vamos a modificar la configuración para poder acceder a esta web desde cualquier IP.

¡Cuidado!

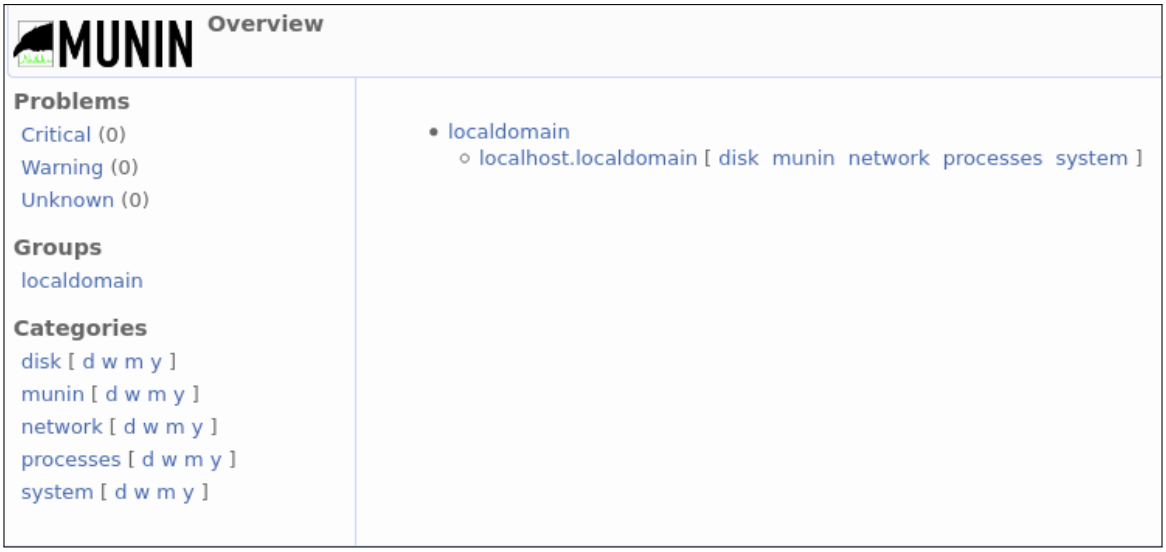
Permitir el acceso desde cualquier IP es un fallo de seguridad. Habría que modificar la configuración y poner autenticación con "auth-basic".

Para ello, modificaremos el fichero arriba mencionado y modificaremos las líneas donde aparece "Require" y pondremos "**Require all granted**". Tras realizar la modificación de la configuración de Apache, deberemos reiniciar el servicio:

Reiniciamos Apache para que coja la configuración

```
root@vega:~# systemctl restart apache2
```

Y tras esto, ya deberíamos poder ir al navegador web y poniendo **http://IPSERVIDOR/munin** (donde IPSERVIDOR es la IP del servidor que acabamos de configurar) y ver el interfaz de Munin:



Interfaz de Munin

6.3. Configuración

Tras la instalación, Munin por defecto tiene una configuración sencilla que hace una monitorización básica del servidor. La configuración se puede encontrar en `/etc/munin/` y podemos diferenciar 3

ficheros/directorios:

- **munin.conf**: fichero de configuración principal. En él se puede configurar todo lo relacionado con Munin, avisos, procesos...
- **munin-node.conf**: es el fichero de configuración que se utiliza si se utiliza el sistema básico de monitorización centralizada. En este fichero especificaremos desde qué servidores permitiremos que puedan monitorizar el propio servidor.
- **plugins**: directorio con los plugins activados que se van a utilizar durante la monitorización. En este directorio están los enlaces simbólicos de los plugins que van a ser utilizados durante la monitorización del servidor. Los plugins están en `/usr/share/munin/plugins/`. Normalmente suelen ser scripts en los lenguajes Perl, Python o Shell.

6.4. CRON

La monitorización del servidor se realiza a través de un proceso CRON que instala Munin y que por defecto se hace cada 5 minutos. El fichero CRON está situado en `/etc/cron.d/munin`, y cuenta con 4 procesos automáticos que se ejecutan en el servidor:

- Proceso de monitorización
- 3 procesos de limpieza de caché y ficheros temporales que son creados durante la monitorización

7. Virtualbox y adaptadores de red

7.1. Introducción

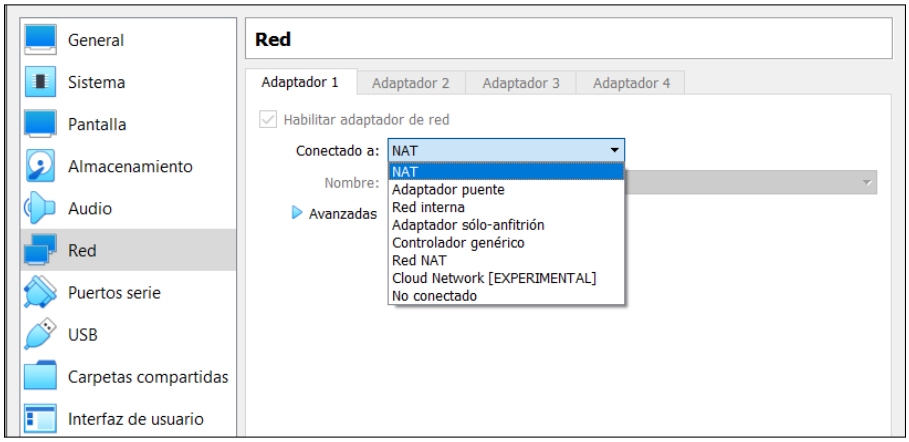
Virtualbox es una herramienta de virtualización para crear máquinas virtuales de manera sencilla. Es multiplataforma por lo que se puede utilizar en Windows, MacOS y Linux y aparte, es Software Libre.

Este documento no va a entrar en detalle en cómo se crean las máquinas virtuales, sino que va a explicar los distintos modos y adaptadores de red que puede tener una máquina virtual en este sistema de virtualización.

7.2. Adaptadores de red

Virtualbox permite que cada máquina virtual cuente con hasta cuatro adaptadores de red, lo que comúnmente se llaman interfaces o NIC (network interface controller).

Al crear las máquinas virtuales sólo tienen un único adaptador activo y suele estar configurado en modo NAT, pero tal como se ve a continuación, en el desplegable se puede ver que existen otras opciones:



En la documentación oficial aparece la explicación de los distintos modos, y es buena práctica leer y entender la documentación del software que utilizamos. También hay que entender que cada tipo de adaptador contará con una serie de ventajas y una serie de limitaciones que aparecen reflejadas en la documentación. Estos modos son comunes a otros sistemas de virtualización (VmWare, Proxmox, …), pero el nombre o el modo de uso puede variar así como las posibles limitaciones que puedan existir.

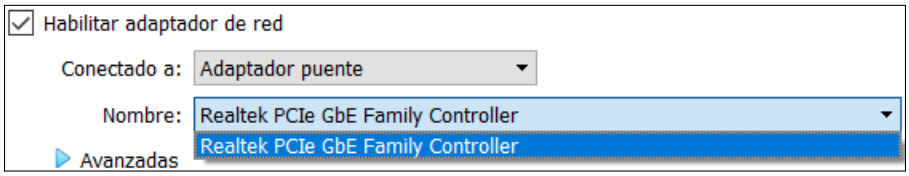
A continuación se va a dar una pequeña introducción a cada tipo de adaptador.

7.2.1. Adaptador puente

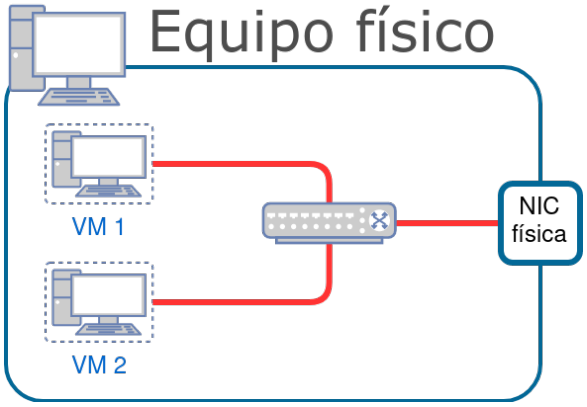
Es el tipo de adaptador que se usará si queremos que las máquinas virtuales aparezcan en la red física como si fueran un equipo más. Para poder entenderlo de mejor manera, podríamos pensar que este tipo de adaptador lo que hace es crear un “switch virtual” entre las máquinas virtuales y el interfaz físico, por lo que es como si fueran un equipo más en la red física.

Si el equipo físico anfitrión cuenta con más de un NIC (por ejemplo, en un portátil el NIC por cable y el NIC

wifi) tendremos que elegir en la máquina virtual sobre qué NIC queremos hacer el puente. En la siguiente imagen en el desplegable sólo se puede seleccionar un interfaz porque el equipo sólo cuenta con un NIC físico.



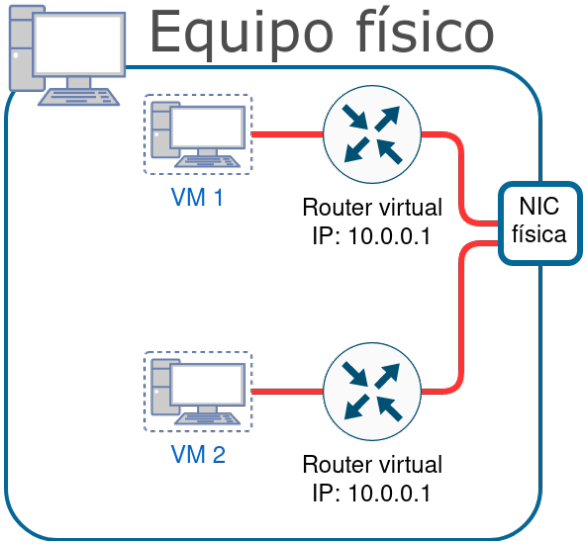
Es el método utilizado cuando virtualizamos servidores, ya que podrán dar sus servicios a toda la red.



Red como Adaptador Puente

7.2.2. NAT

Cada máquina virtual contará con su propio “router virtual” que hará NAT, y por eso todas las máquinas virtuales que usen este modo suelen tener la misma IP, pero no pertenecen a la misma red. Por defecto no se puede realizar conexión desde la red física al equipo virtualizado.



Red en modo NAT

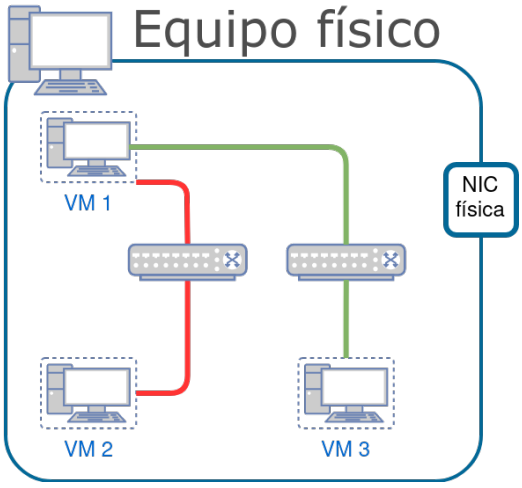
7.2.3. Red interna

Este tipo de adaptador lo que hace es “crear” un “switch virtual” que unirá las distintas máquinas virtuales que estén conectadas al nombre de esa red interna.

En el siguiente ejemplo la VM1 tiene 2 NICs, cada una con una red interna distinta. La VM2 tiene un NIC conectado a una de las redes internas creadas previamente y VM3 está conectada a la otra red interna.

Virtualbox no se encarga de dar IPs en estas redes, por lo que deberemos configurar cada interfaz de la máquina virtual con el direccionamiento que nos interese.

Este método se utiliza si queremos comunicar máquinas virtuales entre sí y que estén aisladas, ya que no podrán conectarse con el exterior, ni siquiera con el propio equipo físico.

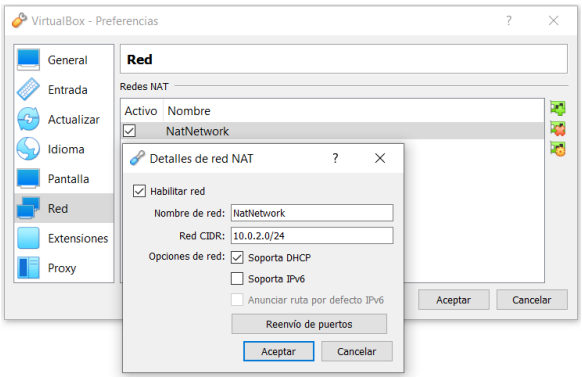


Red Interna

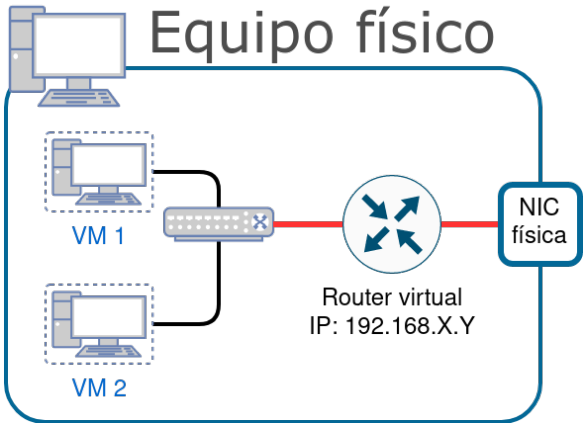
7.2.4. Red NAT

Podría definirse como una mezcla de NAT y red interna. Las máquinas virtuales podrán pertenecer a una única red, se podrán comunicar entre ellas, estarán detrás de un NAT de la red física y se podrán comunicar con el exterior.

Para poder usar ese modo hay que crear la “red NAT” en Virtualbox yendo a “*Archivo → Preferencias → Red*” y ahí se creará las redes NAT que queramos con el direccionamiento interno que nos interese.



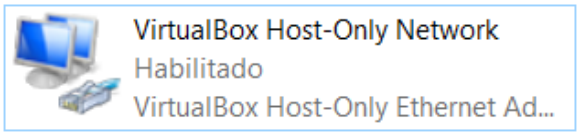
A la hora de crear la máquina virtual y elegir la opción “Red NAT” se podrá elegir entre las redes creadas en el paso anterior.



Red NAT

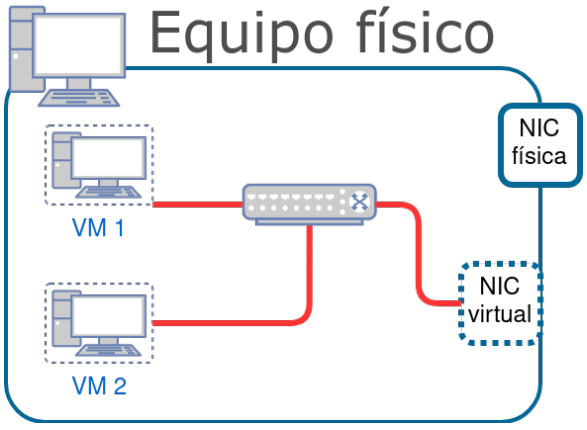
7.2.5. Adaptador sólo-anfitrión

Este tipo de adaptador es similar al de "red interna" pero con la posibilidad de comunicarse con el equipo físico anfitrión. En el equipo físico se crea un interfaz virtual y a través de él se podrá comunicar con las máquinas virtuales.



El direccionamiento que existe entre las VMs y el host se define en Virtualbox, dentro de "*Archivo* → ***Administrador de red de anfitrión***". Las máquinas virtuales podrán coger IP de ese direccionamiento por DHCP.

Mismo uso que "red interna" pero añadiendo la opción de comunicarnos con el host anfitrión.



Red en modo "Sólo Anfitrión"

7.3. Resumen de los adaptadores

A continuación se expone una tabla que resume los distintos tipos de adaptadores que existen y la conectividad posible entre las máquinas virtuales que usan esos adaptadores y el host anfitrión (fuente).

Modo \ Conectividad	VM → Host	VM ← Host	VM1 ← → VM2	VM → LAN real	VM ← LAN real
Adaptador puente	✔	✔	✔	✔	✔
NAT	✔	Redirección de puertos	✘	✔	Redirección de puertos
Red interna	✘	✘	✔	✘	✘
Sólo-anfitrión	✔	✔	✔	✘	✘

En la documentación se explica cómo realizar la redirección de puertos.