

Diseño e implementación de una red informática y sistema de monitorización de Servicios Windows

Diego Prieto

2 ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED

Quiero agradecer de manera especial a mi padre, quien me ha brindado su apoyo y asesoramiento durante todo el proceso de desarrollo de este proyecto. Sus consejos y conocimientos fueron de gran ayuda y contribuyeron significativamente en la consecución de los objetivos planteados. También quiero agradecer a Solitium por haber prestado generosamente el servidor necesario para realizar las pruebas y ejecutar el proyecto. Su colaboración fue fundamental para poder llevar adelante este proyecto.

Índice

Introducción	2
Objetivos	3
1. Distribución del Proyecto	4
2. Hardware.....	5
3. Diseño de la Red.....	6
4. Creación del entorno virtual.....	8
4.1. Acceso al SO VMware ESXi	8
4.2. Creación de VM con VMware ESXi	9
4.3. Instalación de los SO	11
4.3.1. IPfire	11
4.3.2. Ubuntu Server	14
4.3.3. Ubuntu Desktop	16
4.3.4. Windows Server	17
5. Comunicación entre redes	18
5.1. Configuración de Enrutador	18
5.2. Configuración de Firewalls	19
6. Instalación y Configuración de Servicios	20
6.1. DHCP.....	20
6.2. Servidor Impresión	23
6.3. Servidor de archivos.....	24
6.4. DNS.....	25
6.5. AD.....	26
6.6. XAMPP	28
6.7. FilleZilla Server	29
6.8. Replicación	30
6.8.1. Replicación AD.....	30
6.8.2. Replicación MySQL	32
7. Creación de bases de datos SQL.....	33
7.1. Creación general de una BBDD.....	33
7.2. BBDD INFO_AD	34
7.3. BBDD INFO_CARPETAS_COMPARTIDAS	35
7.4. BBDD INFO_DHCP	36
7.5. BBDD INFO_DNS.....	38
7.6. BBDD INFO_EQUIPOS	38
7.7. BBDD INFO_IMPRESORAS	39

8.	Logs BBDD	41
8.1.	Ruta y formato de logs	41
8.2.	Creación de procedimientos	41
8.3.	Acceso a logs	42
9.	Creación de Scripts Powershell	43
9.1.	SQL dentro de Powershell	43
9.2.	Función de conexión a la base de datos.....	43
9.3.	Ejecución de SQL a través de Powershell	44
9.4.	Cmdlts utilizados	44
9.4.1.	AD	44
9.4.2.	DHCP.....	45
9.4.3.	Carpetas Compartidas	45
9.4.4.	DNS.....	45
9.4.5.	Servidor Impresión	45
9.4.6.	Sistemas.....	46
10.	Automatización de Scripts PowerShell.....	47
11.	Creación de la web	49
11.1.	Página principal	49
11.2.	Páginas secundarias	54
	Conclusión	56
	Acceso a los recursos del proyecto	56
	Bibliografía	57

Introducción

En este proyecto se ha diseñado e implementado una red de servidores dividida en dos sites; cada uno de ellos con una configuración específica y una función de proveer servicios específicos. Además, se han creado una serie de scripts de PowerShell para recolectar información de diferentes servicios de Windows y almacenarlos en una base de datos MySQL. La información recolectada se ha integrado en una página web para su acceso y análisis. A lo largo de este proyecto se detallará el proceso de diseño e implementación de esta red de servidores, así como la creación y automatización de los scripts de PowerShell para la recolección de datos y su almacenamiento en la BBDD. También se explicará el proceso de integración de la información en una página web para su análisis y acceso.

Objetivos

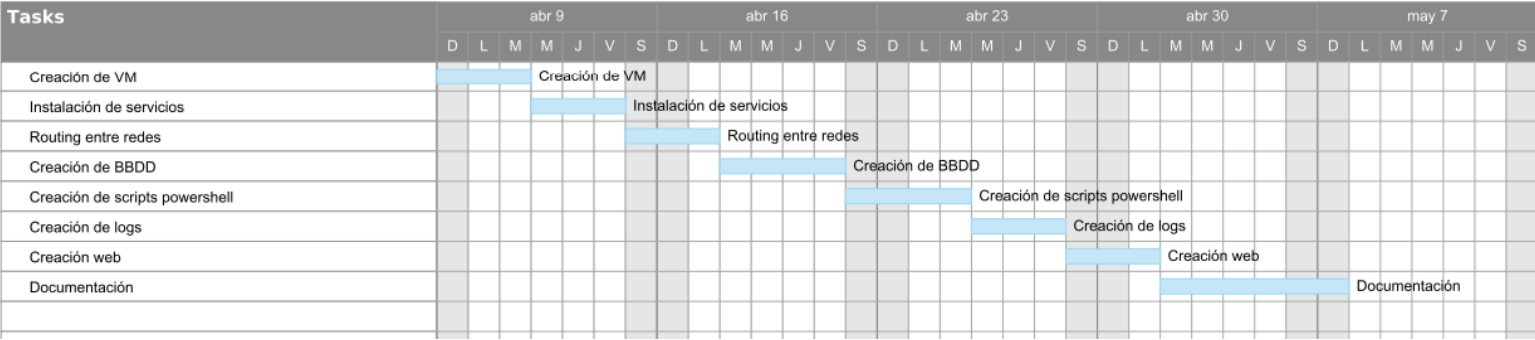
A continuación, se presenta una lista de objetivos del proyecto:

- Diseñar una red de servidores dividida en dos sites conectados a través de un enrutador y firewalls de tres patas.
- Configurar diferentes servicios en los servidores de ambos sites, incluyendo AD, DNS, DHCP, carpetas compartidas, servidor de impresión, MySQL, FTP y Apache.
- Crear diferentes scripts de PowerShell para tomar información de los diferentes servicios windows en los servidores y enviarlos a una base de datos MySQL.
- Crear diferentes procedimientos SQL para la creación de logs de la información de los diferentes servicios windows en la base de datos MySQL.
- Crear una página web que acceda a la información almacenada en la base de datos MySQL y presente de una manera clara y accesible.
- Automatizar la ejecución de los scripts de PowerShell utilizando el programador de tareas de Windows.
- Garantizar la seguridad de la red y los servidores mediante la configuración adecuada de los firewalls y el enrutador.
- Realizar pruebas exhaustivas de la red y los servicios para garantizar su correcto funcionamiento.
- Documentar detalladamente todo el proceso de diseño, configuración y prueba de la red y los servicios.

1. Distribución del Proyecto

En este apartado realizamos un diagrama de Gantt que muestra la planificación y distribución de tiempo del proyecto. Este diagrama es una herramienta útil para visualizar las diferentes tareas que se deben completar y la duración estimada de cada una de ellas. Además, nos permite establecer un cronograma y un orden de prioridades para cumplir con los objetivos del proyecto en el tiempo previsto.

A continuación, se muestra el diagrama de Gantt:



2. Hardware

En el proyecto se ha utilizado un servidor PowerEdge R720 de Dell, que es un servidor de alta gama diseñado para satisfacer las demandas de cargas de trabajo intensivas. Cuenta con una memoria RAM de 64 GB, lo que permite una rápida ejecución de aplicaciones y un procesamiento eficiente de grandes cantidades de datos. Además, se han instalado dos discos duros en el servidor, uno de 6 TB y otro de 7 TB, lo que proporciona una gran capacidad de almacenamiento para los datos y recursos del proyecto.

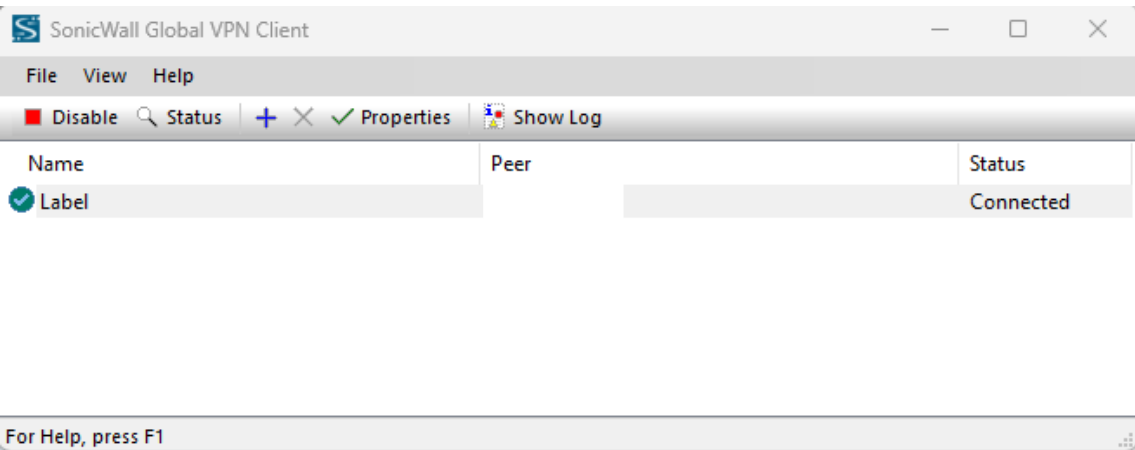
El servidor se ha instalado con el sistema operativo DellEMC-VMware ESXi, que es un hipervisor que permite la virtualización de servidores y la creación de múltiples máquinas virtuales en un único servidor físico. Esto proporciona una mayor flexibilidad y escalabilidad en la gestión de los recursos de la red del proyecto. El sistema operativo ESXi también cuenta con características de seguridad avanzadas que protegen los datos y recursos de la red de posibles amenazas.

En resumen, el servidor PowerEdge R720 de Dell con 64GB de RAM, dos discos duros de gran capacidad y el sistema operativo DellEMC-VMware ESXi, es una solución robusta y escalable que proporciona una alta disponibilidad de recursos y una gestión eficiente y segura de los datos y aplicaciones críticos del proyecto.

Estado de la alimentación	ENCENDIDO
Modelo del sistema	PowerEdge R720
Revisión del sistema	1
Nombre de host del sistema	localhost
Sistema operativo	DellEMC-VMware ESXi
Versión del sistema operativo	6.7.0 Update 3 Build-17700523 (A15)
Etiqueta de servicio	4ZCRS22
Código de servicio rápido	10844896970
Versión del BIOS	2.9.0
Versión del firmware	2.65.65.65
Direcciones IP	10.10.25.222
Dirección MAC del iDRAC	B0:83:FE:C1:EC:C9
Nombre del dominio DNS	labelordenadores.local
Firmware de Lifecycle Controller	2.65.65.65
Versión de firmware de iDSDM	N/A
Ubicación	
Licencias	<input checked="" type="checkbox"/> Enterprise



A este servidor se accede mediante una VPN configurada en la red donde está alojado el servidor usando el cliente de la empresa SonicWall.



3. Diseño de la Red

Como primer paso en el diseño de una red es importante considerar varios aspectos para garantizar su eficiencia y seguridad, desde la topología, los protocolos de comunicación, la selección de hardware y software, hasta la gestión y monitoreo de la red. Un buen diseño de red debe permitir una comunicación fluida entre los diferentes dispositivos y usuarios, asegurando al mismo tiempo la privacidad y protección de la información transmitida. En este apartado, se presentará el diseño de la red creada para satisfacer las necesidades de la organización y los objetivos planteados.

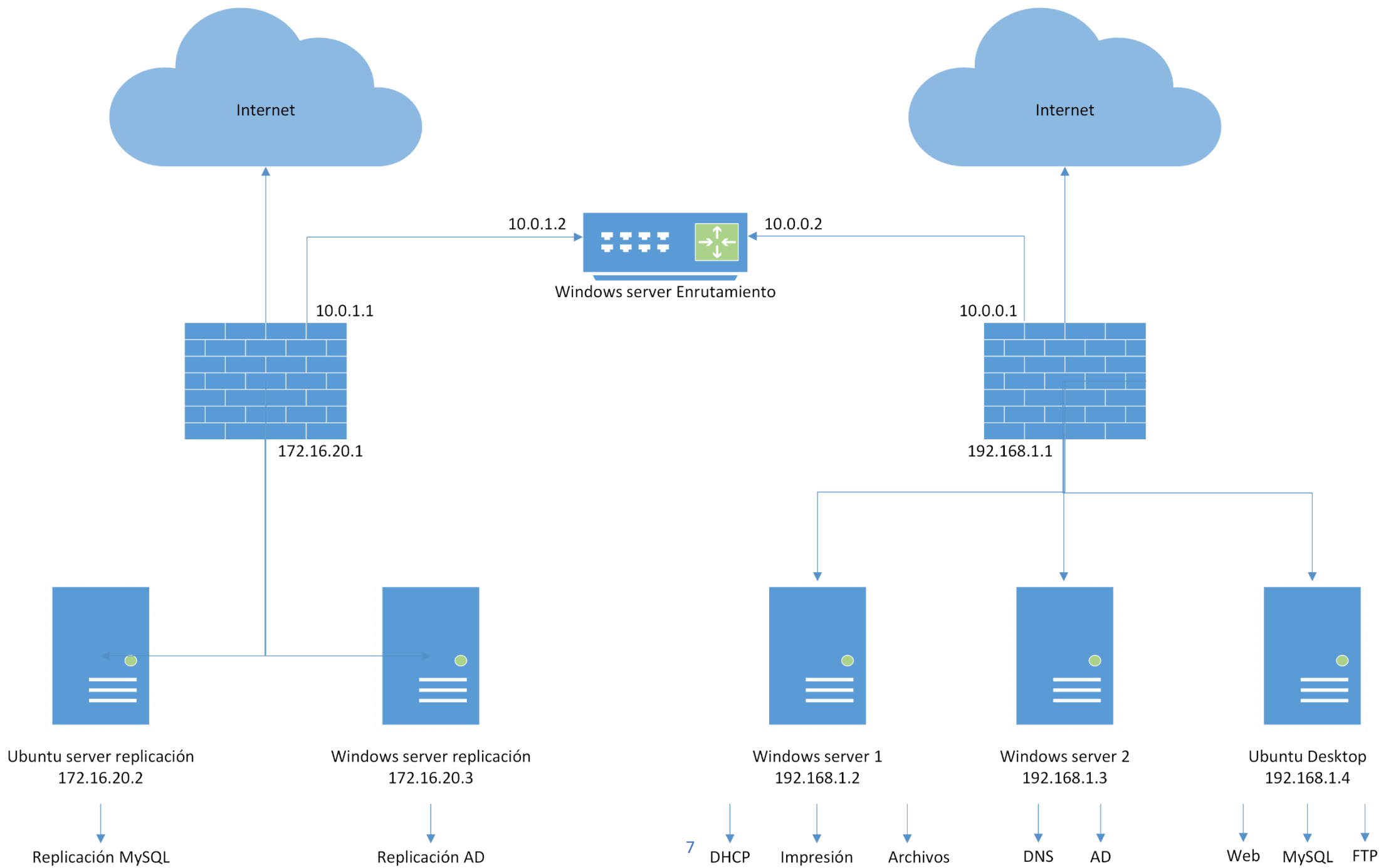
El diseño de la red consta de dos sites, cada uno de ellos con un firewall de tres patas, una para la red LAN del site, otra para internet y la última para conectar los firewalls entre sí con un enrutador que será un servidor Windows. El objetivo de esta configuración es crear una red segura y eficiente que permita la interconexión de los dos sites.

En el primer site se encuentran dos servidores Windows que tendrán diferentes funciones. Uno de ellos actuará como DNS, DHCP, servidor de carpetas compartidas, servidor de impresión y AD. El otro servidor se dedicará a otras funciones, como la gestión de bases de datos con MySQL y la administración de un servidor web con Apache.

En el segundo site se ha configurado un servidor Windows para la replicación de AD y un servidor Ubuntu para la replicación de MySQL. Esto permite una mayor redundancia y tolerancia a fallos en caso de que uno de los servidores falle o se desconecte de la red.

El diseño de la red tiene como objetivo proporcionar una infraestructura segura y eficiente para la comunicación entre los dos sites, con servidores dedicados para diferentes funciones y con mecanismos de redundancia para asegurar la disponibilidad de los servicios críticos. La configuración de los firewalls y el enrutador proporciona una capa adicional de seguridad para proteger la red de posibles amenazas externas.

A continuación, se presenta un esquema visual de la red descrita anteriormente para facilitar su comprensión y visualización:

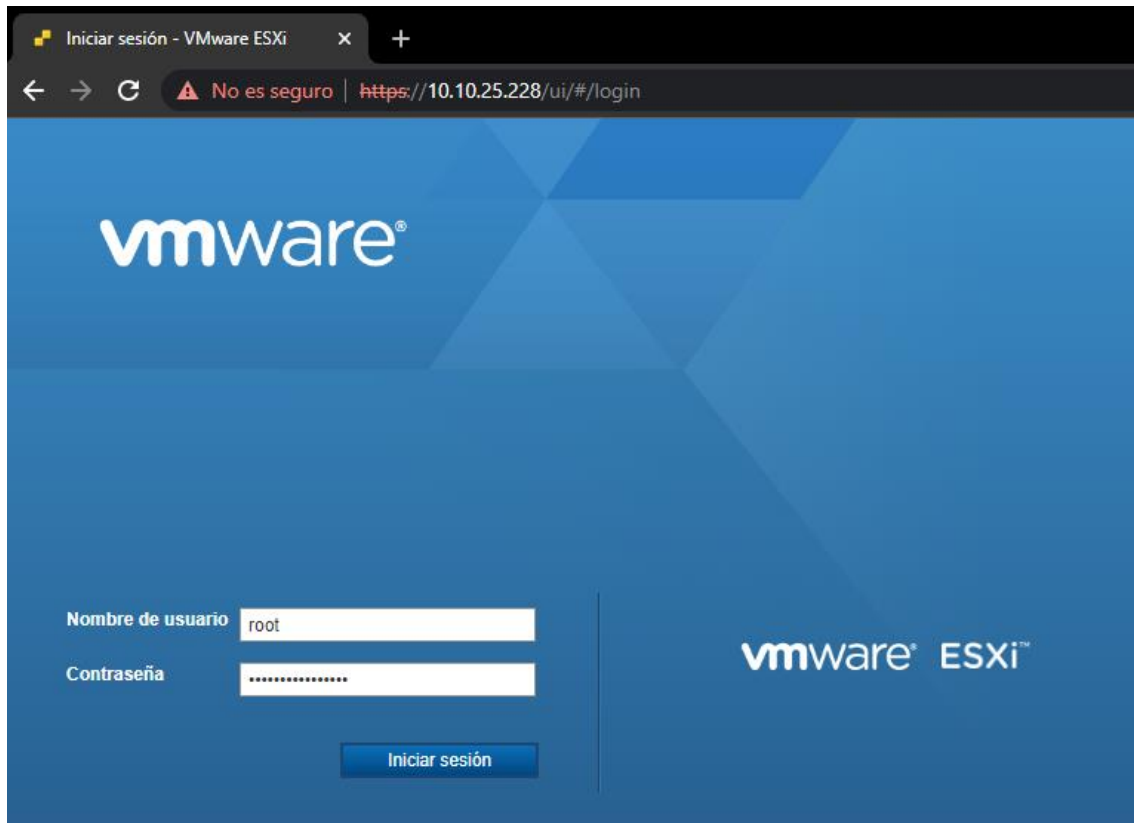


4. Creación del entorno virtual

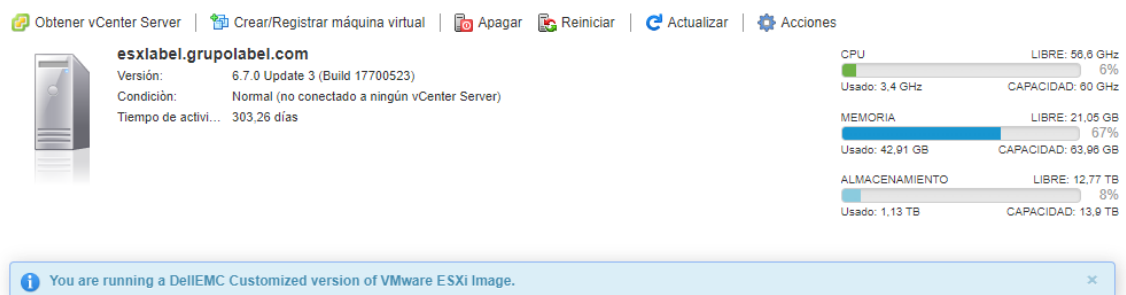
En este apartado se va a desarrollar como creamos las VM e instalamos sus respectivos SO para la realización del proyecto.

4.1. Acceso al SO VMware ESXi

Para poder acceder al SO y crear las diferentes VM debemos acceder al servidor mediante un navegador introduciendo la IP previamente configurada en el servidor físico. En mi caso la IP es: 10.10.25.228

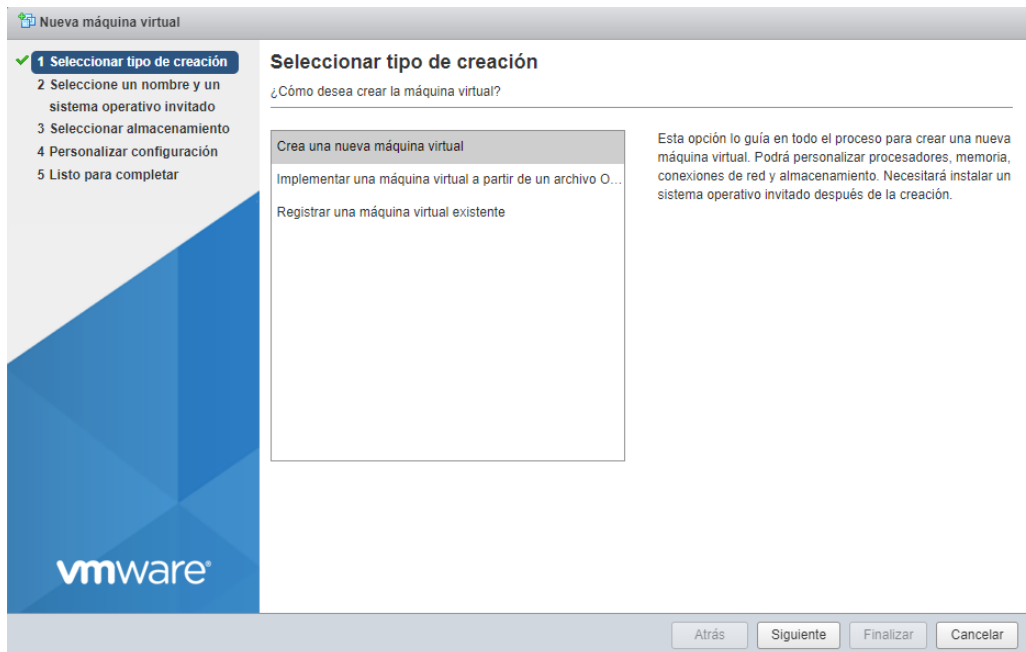


Una vez hemos accedido al sistema con nuestro usuario y contraseña podemos crear las diferentes VM haciendo clic en la opción: “Crear/Registrar máquina virtual”



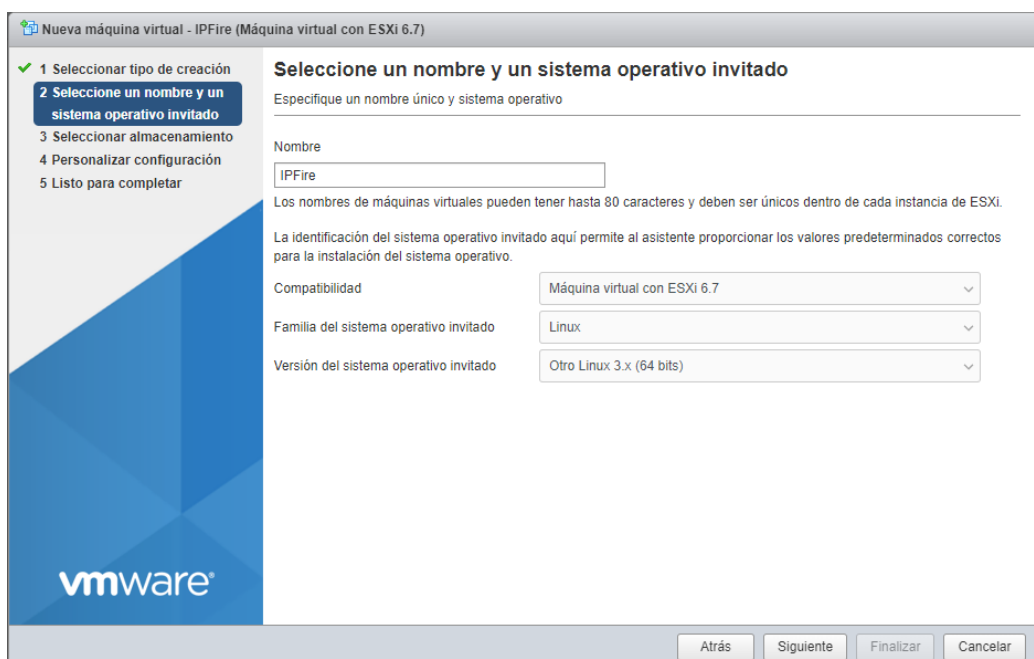
4.2. Creación de VM con VMware ESXi

Para la instalación de los diferentes SO que usaremos a lo largo del proyecto debemos generar una serie de VM para alojar los sistemas. Una vez hemos accedido al wizard de creación de VM veremos una pantalla como ésta:



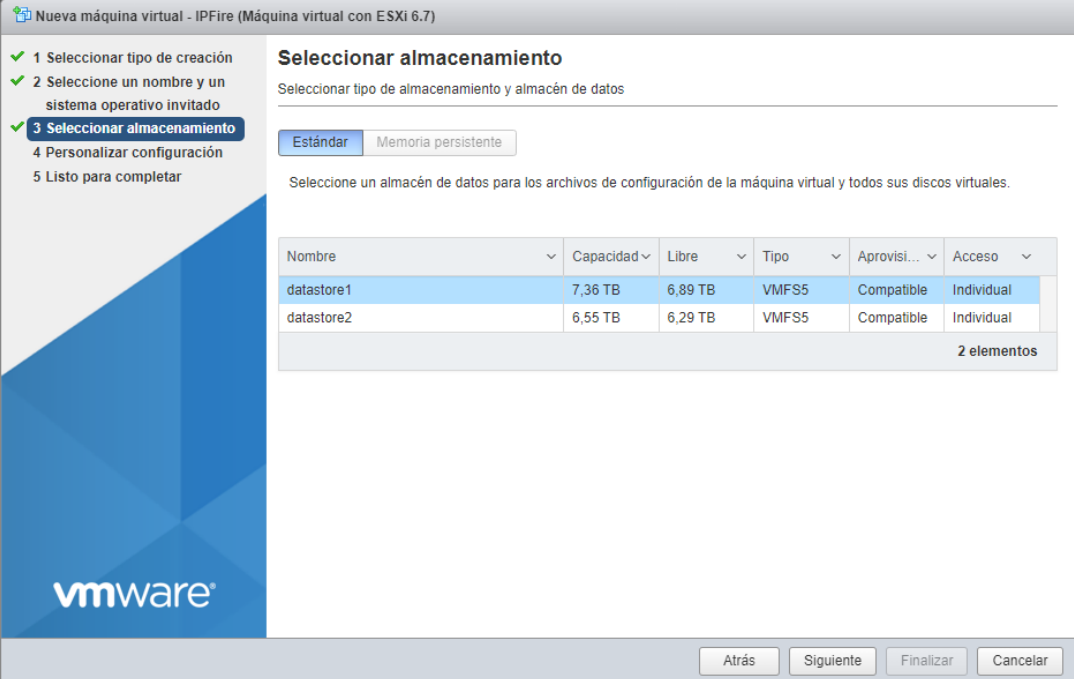
En esta pantalla podemos seleccionar lo que queremos hacer, podemos crear una VM desde cero, registrar una máquina desde un archivo OVA o registrar una máquina ya existente en uno de nuestros DataStorage. Para empezar a crear las máquinas elegimos “Crear una nueva máquina virtual” y pulsamos siguiente.

En la siguiente pantalla debemos de darle un nombre a nuestra VM y seleccionar tanto la familia del SO que vamos a instalar como su versión.



En este caso vamos a configurar uno de los firewalls virtuales que vamos a necesitar. Le asignamos el nombre y su sistema operativo y pulsamos siguiente.

Ahora nos pregunta dónde queremos guardar nuestra VM.



Nueva máquina virtual - IPFire (Máquina virtual con ESXi 6.7)

- ✓ 1 Seleccionar tipo de creación
- ✓ 2 Seleccione un nombre y un sistema operativo invitado
- ✓ 3 Seleccionar almacenamiento
- 4 Personalizar configuración
- 5 Listo para completar

Seleccionar almacenamiento

Seleccionar tipo de almacenamiento y almacén de datos

☐ Estándar ☐ Memoria persistente

Seleccione un almacén de datos para los archivos de configuración de la máquina virtual y todos sus discos virtuales.

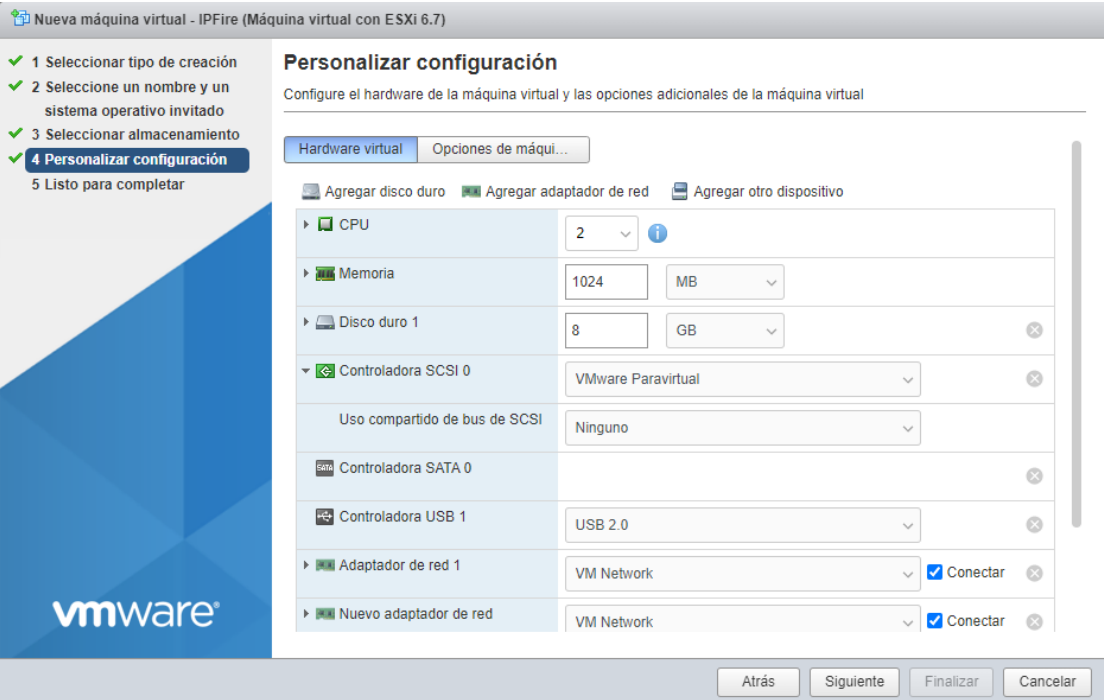
Nombre	Capacidad	Libre	Tipo	Aprovisi...	Acceso
datastore1	7,36 TB	6,89 TB	VMFS5	Compatible	Individual
datastore2	6,55 TB	6,29 TB	VMFS5	Compatible	Individual

2 elementos

Atrás Siguiente Finalizar Cancelar

Debemos elegir la ubicación y presionar siguiente.

En la última pantalla podemos modificar los recursos que se asignarán a la VM, el sistema coloca unos recursos predeterminados en función de la versión de SO que hemos elegido previamente. También podemos añadir diferentes tarjetas de red o discos más discos, así como seleccionar la ISO que se va a usar para instalar el sistema.



Nueva máquina virtual - IPFire (Máquina virtual con ESXi 6.7)

- ✓ 1 Seleccionar tipo de creación
- ✓ 2 Seleccione un nombre y un sistema operativo invitado
- ✓ 3 Seleccionar almacenamiento
- ✓ 4 Personalizar configuración
- 5 Listo para completar

Personalizar configuración

Configure el hardware de la máquina virtual y las opciones adicionales de la máquina virtual

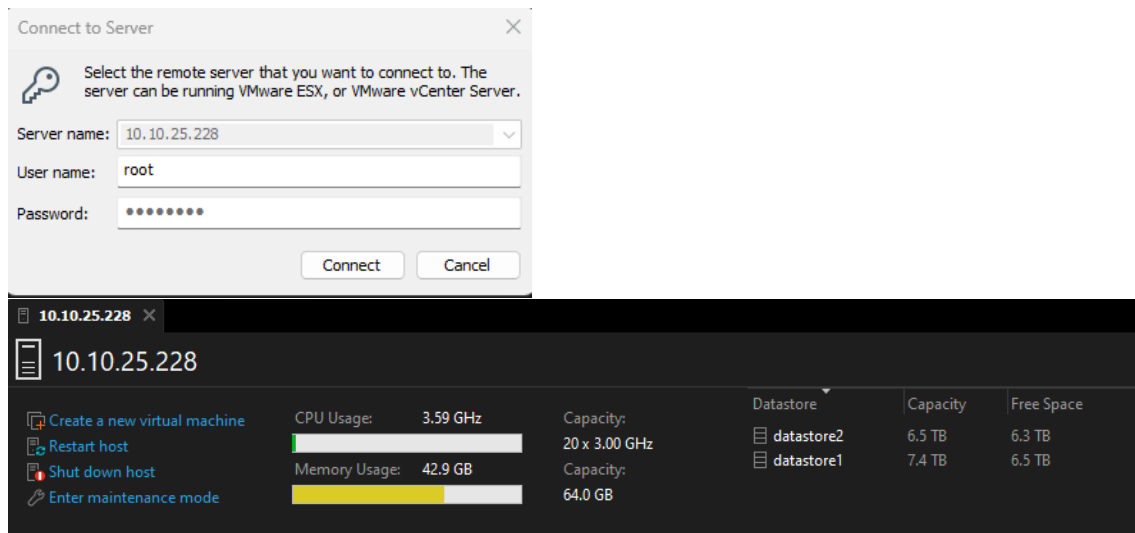
☐ Hardware virtual ☐ Opciones de máqui...

CPU	2	
Memoria	1024	MB
Disco duro 1	8	GB
Controladora SCSI 0	VMware Paravirtual	
Uso compartido de bus de SCSI	Ninguno	
Controladora SATA 0		
Controladora USB 1	USB 2.0	
Adaptador de red 1	VM Network	<input checked="" type="checkbox"/> Conectar
Nuevo adaptador de red	VM Network	<input checked="" type="checkbox"/> Conectar

Atrás Siguiente Finalizar Cancelar

Este proceso se tiene que realizar cada vez que queramos crear una nueva VM. De forma adicional se pueden crear plantillas para automatizar la creación de estas VM.

Una vez se han creado las diferentes VM podemos acceder a ellas desde el propio navegador o utilizar el programa: *VMware Workstation Pro* para acceder a las diferentes VM usando la IP del servidor y el mismo usuario y contraseña que hemos usado para entrar en la plataforma ESXi.



4.3. Instalación de los SO

Una vez hemos creado las diferentes máquinas y asignado las diferentes ISO debemos instalar los diferentes sistemas. Aquí solo se explicará un ejemplo de instalación de cada versión de SO que se va a usar ya que la instalación de dos SO de la misma versión es exactamente igual. Si hubiera alguna diferencia se indicará en el ejemplo.

4.3.1. IPfire

El primer sistema que vamos a instalar es IPfire que es un sistema operativo de firewall de código abierto basado en Linux. Proporciona una amplia gama de funciones de seguridad de red para proteger los sistemas informáticos contra amenazas como virus, malware, intrusiones, ataques de red y otros peligros en línea.

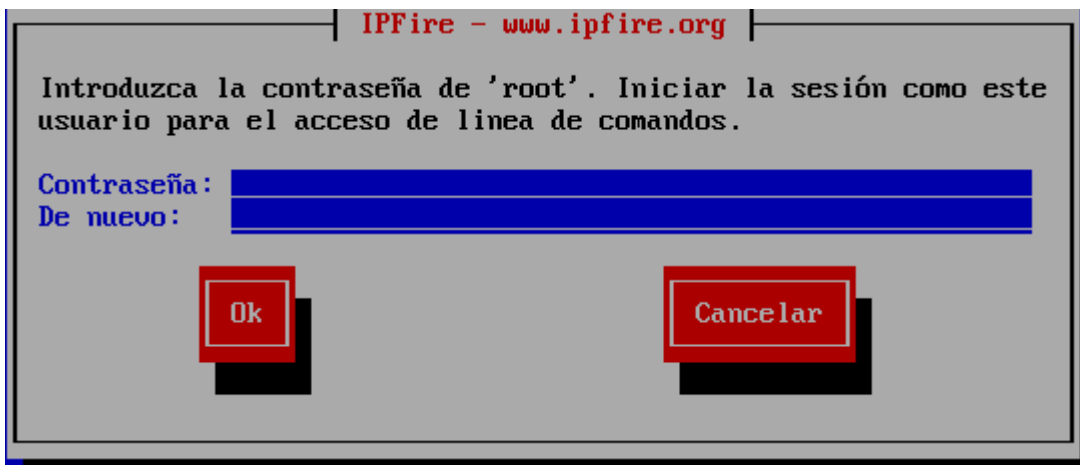
Lo primero que debemos seleccionar es que sistema de ficheros deseamos usar, en este caso ext4.



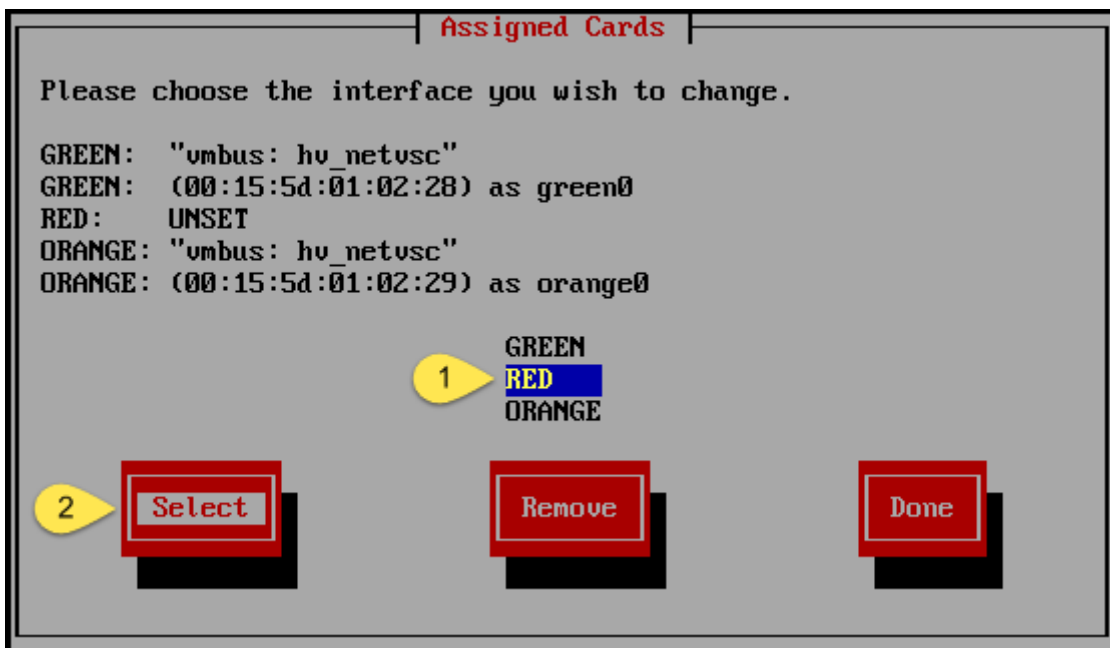
Después debemos asignar un nuevo nombre a la máquina, por defecto asigna el nombre “ipfire”.



Ahora debemos asignar una contraseña al usuario root para que acceda por la línea de comandos. Seguido a esto debemos asignar la contraseña del usuario admin que será el que accederá por la interfaz web.



Seguido a esto debemos asignar que tarjeta de red se asigna a cada una de las diferentes interfaces de red. Nosotros usamos tres (RED + GREEN + ORANGE).



Ahora debemos asignar la IP que tendrá la interfaz GREEN que será la de la red local. En este caso usamos la red 192.168.1.0/24.



Elegimos la IP de la interfaz RED mediante DHCP.



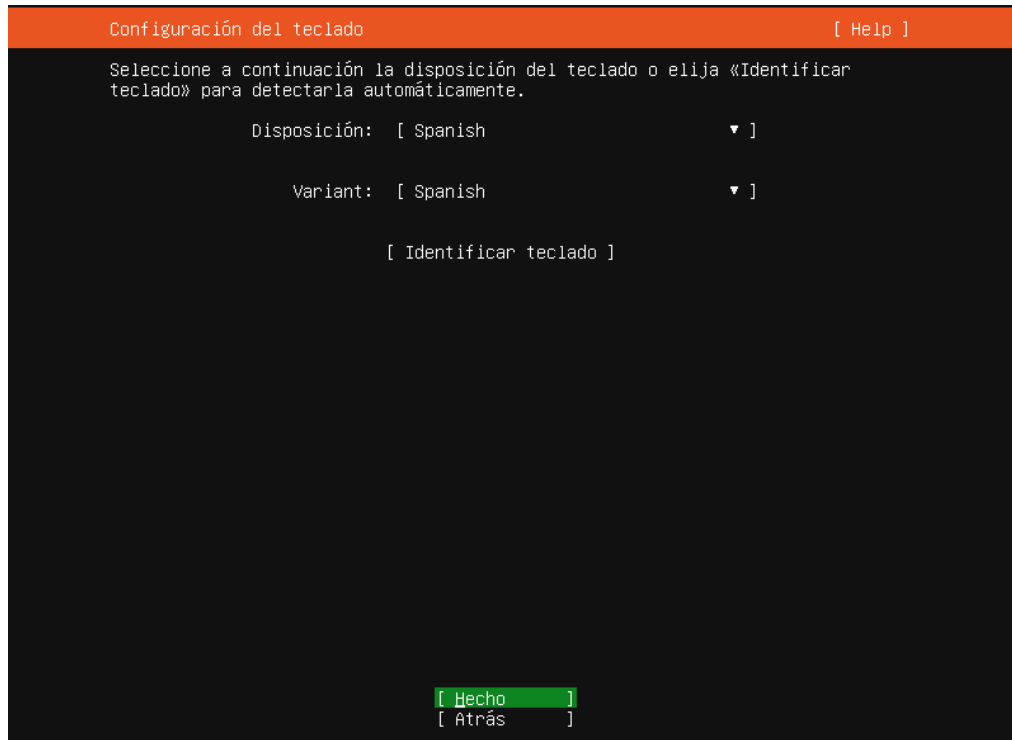
Una vez hecho esto debemos reiniciar el sistema para que se realicen todos los cambios. Una vez dentro del sistema podemos hacer un ping a internet para ver si tenemos conectividad.

```
IPFire-Proyecto login: root
Password:
No mail.
[root@IPFire-Proyecto ~]# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=107 time=7.80 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=107 time=8.24 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
```

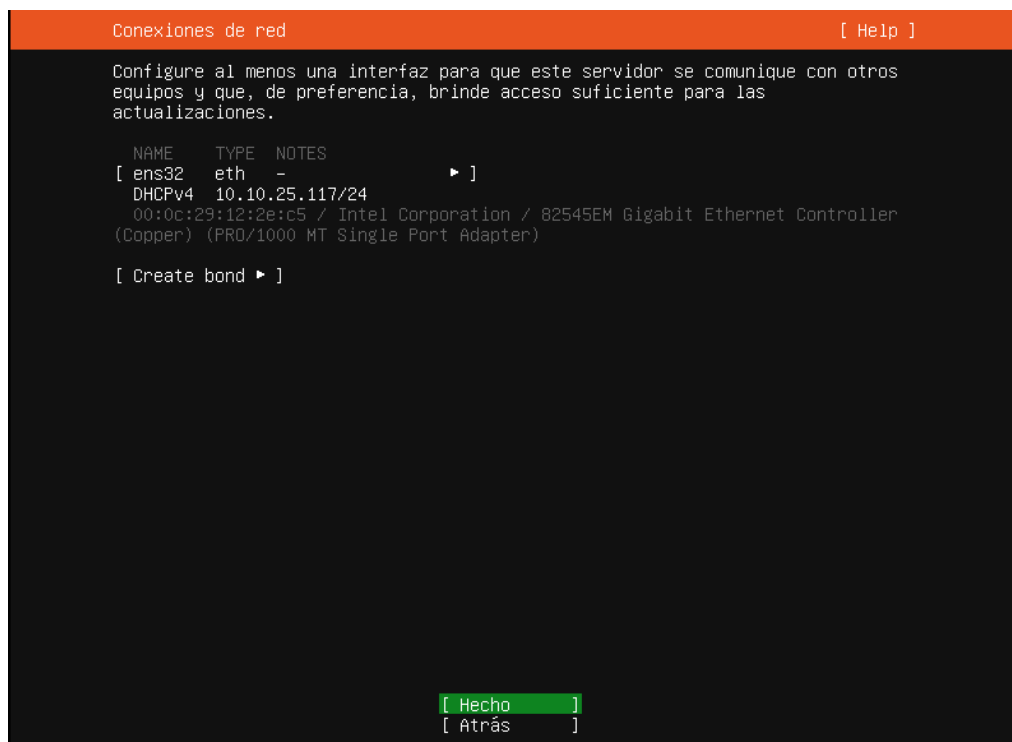
4.3.2. Ubuntu Server

Ahora instalaremos el SO Ubuntu Server que es un sistema operativo de servidor basado en Linux desarrollado por Canonical Ltd. Es una variante de Ubuntu, la distribución de Linux más popular, y está diseñado para ofrecer una plataforma estable, segura y escalable para aplicaciones de servidor.

Primero debemos seleccionar la distribución de teclado que deseamos usar.



Después la conexión de red.



Y por último, el disco que vamos a usar.

Guided storage configuration [Help]

Configure a guided storage layout, or create a custom one:

(X) Use an entire disk

[/dev/sda local disk 16.000G ▼]

[X] Set up this disk as an LVM group

[] Encrypt the LVM group with LUKS

Passphrase:

Confirm passphrase:

() Custom storage layout

[Hecho]

[Atrás]

Ahora creamos el usuario que vamos a usar y procedemos con la instalación.

Profile setup [Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name:

Your server's name:

The name it uses when it talks to other computers.

Pick a username:

Choose a password:

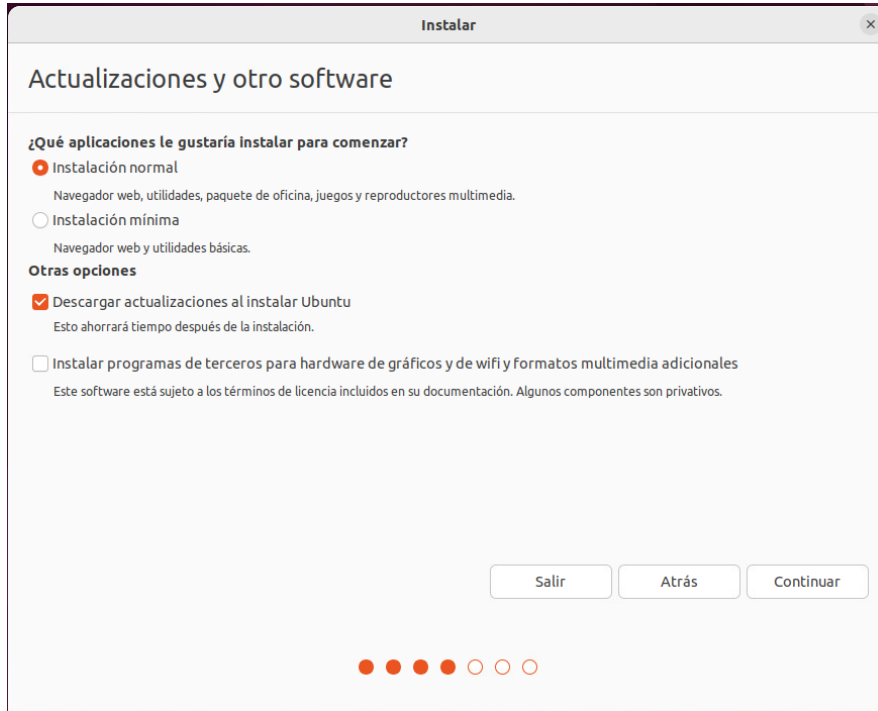
Confirm your password:

[Done] [Back]

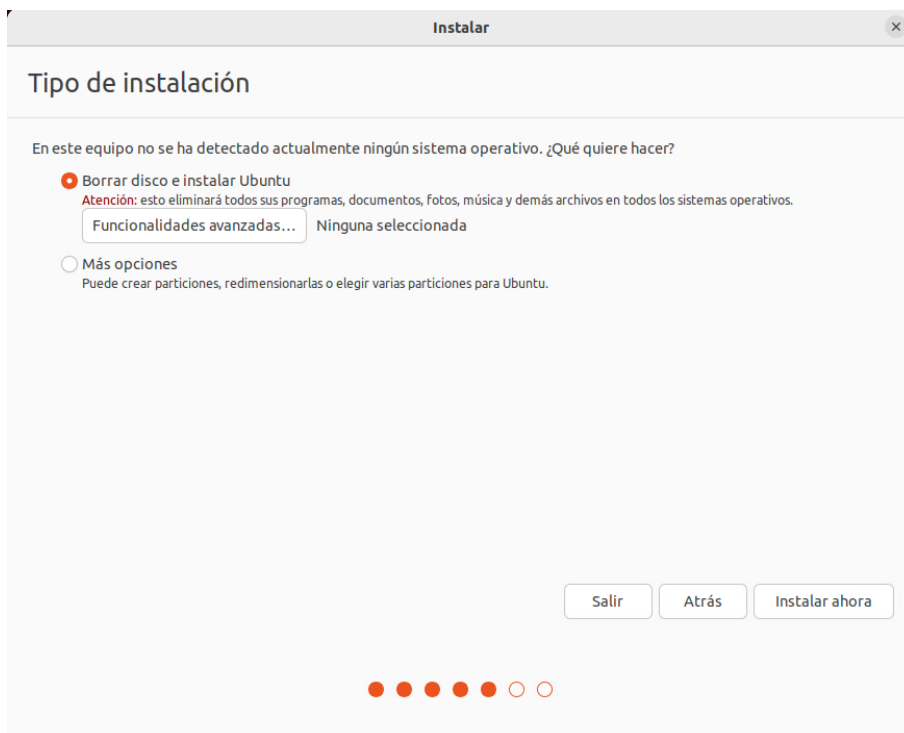
4.3.3. Ubuntu Desktop

Ahora instalaremos el SO Ubuntu Desktop que es un sistema operativo de escritorio basado en Linux desarrollado por Canonical Ltd. Es una variante de Ubuntu, la distribución de Linux más popular, y está diseñado para ofrecer una plataforma estable, segura y fácil de usar para usuarios de escritorio.

Para la instalación del sistema debemos dejar todas las opciones por defecto. Podemos elegir el tipo de instalación que deseamos.



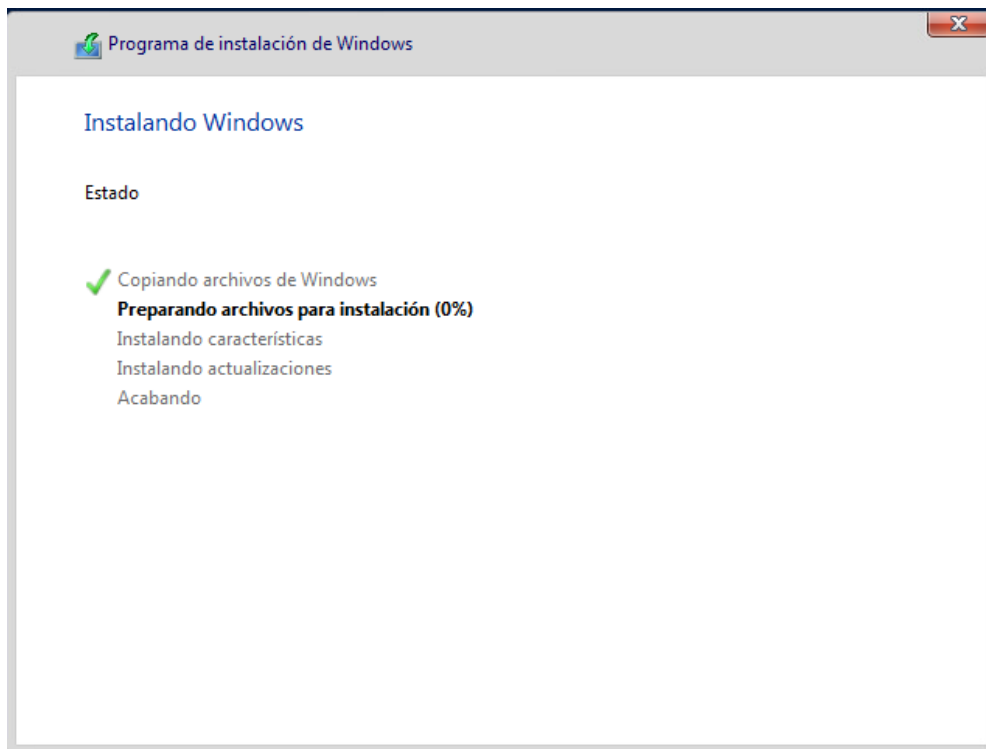
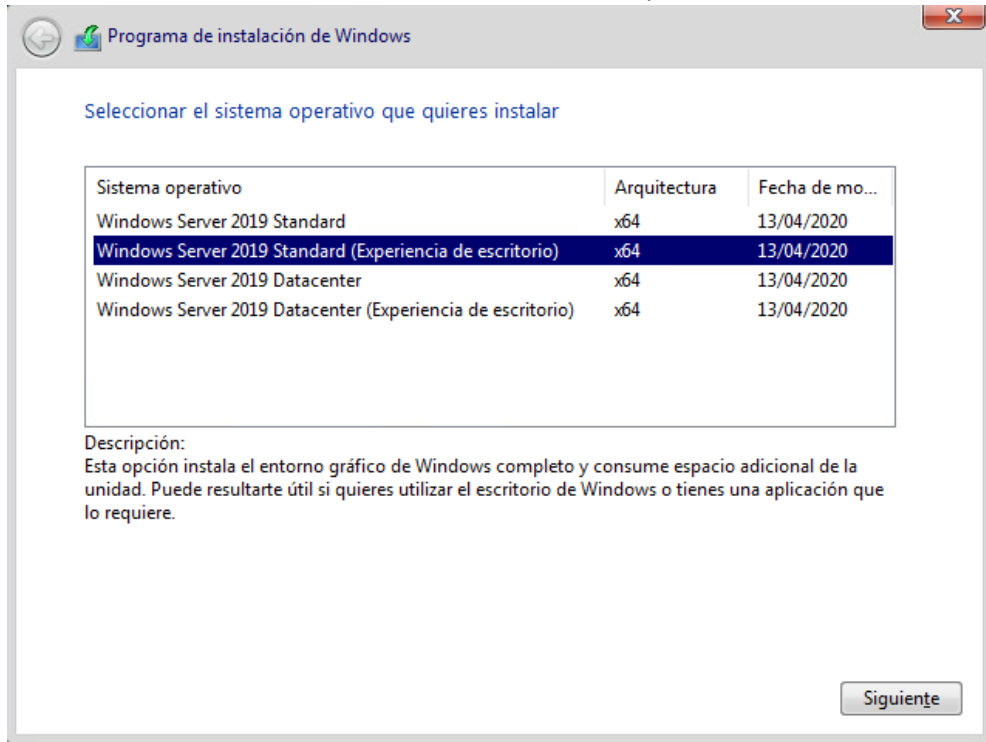
Y luego el disco que queremos usar.



4.3.4. Windows Server

Por último, vamos a instalar el SO Windows server que es un sistema operativo de servidor desarrollado por Microsoft. Está diseñado para proporcionar una plataforma de servidor segura y confiable para ejecutar aplicaciones de servidor en empresas y organizaciones.

Primero debemos de seleccionar la versión que queremos instalar y una vez seleccionado el disco donde se va a instalar aceptamos e instalamos el SO.



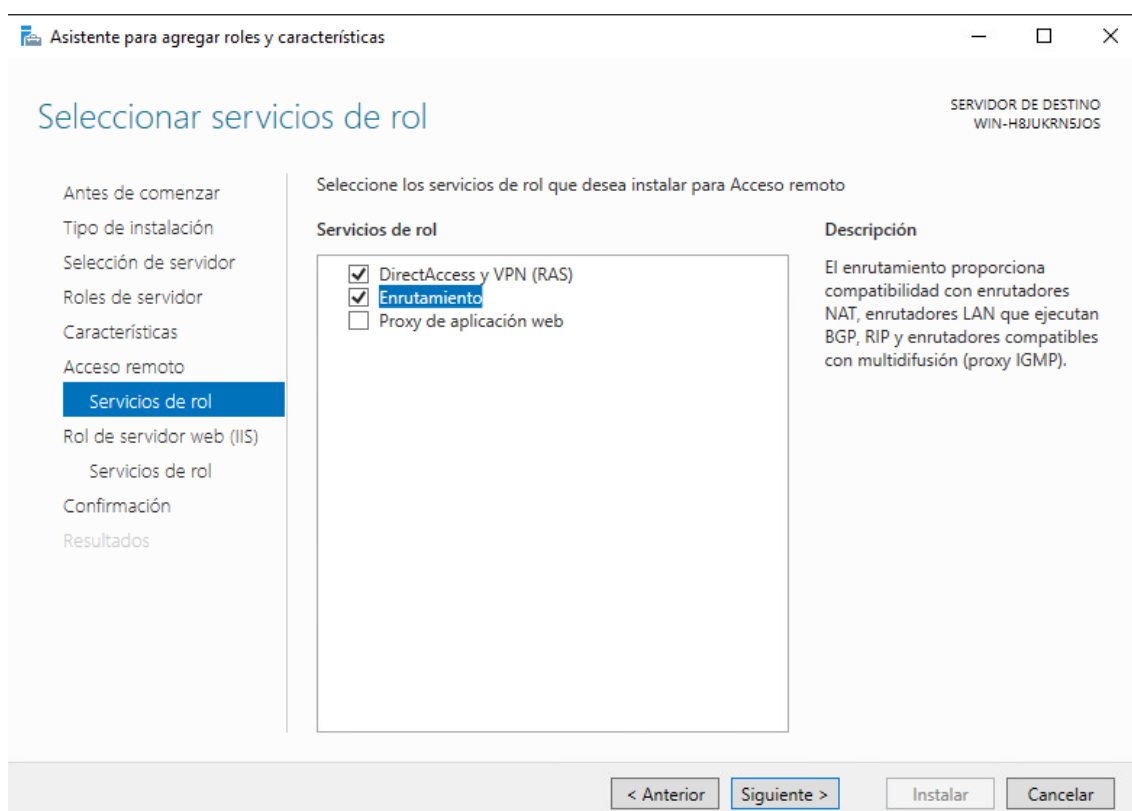
5. Comunicación entre redes

En este apartado se describe la interconexión de dos redes LAN utilizando un router Windows y dos firewalls para mejorar la seguridad y proteger la privacidad de los datos en la red. Se explicará cómo se configuraron los dispositivos de red y se detallará el proceso de interconexión de las redes. La implementación de esta solución permite a los usuarios de ambas redes compartir recursos y datos de manera eficiente y segura.

5.1. Configuración de Enrutador

Para la interconexión de dos redes LAN necesitamos un router que nos permita intercambiar los paquetes entre ambas redes, esto se puede realizar con un router físico o, por ejemplo, con un servidor Windows con el servicio de enrutamiento, que es la opción elegida para este proyecto. Las dos redes LAN utilizadas son 192.168.1.0/24 y 172.16.20.0/24 pero, ya que estas están colocadas bajo un firewall y la conexión entre redes se realiza mediante una red DMZ, debemos de configurar dos nuevas redes que son las que usará el router con dos tarjetas de red para la conexión con los firewalls y ellos se encargan de reenviar los paquetes a ambas LAN. Estas redes serán 10.0.0.0/24 y 10.0.1.0/24.

Para que un servidor Windows lo primero que debemos hacer es instalar el servicio de enrutamiento.



Una vez hemos instalado el servicio asignamos las IP a las diferentes tarjetas de red en este caso: 10.0.0.2 y 10.0.1.2. Hecho esto debemos crear las rutas que ha de seguir el router.

```
C:\Users\Administrador>route add -p 192.168.1.0 mask 255.255.255.0 10.0.0.1
Correcto

C:\Users\Administrador>route add -p 172.16.20.0 mask 255.255.255.0 10.0.1.1
Correcto
```

Para crear las rutas usamos el comando “route add” seguido de “-p” para hacer que la ruta sea persistente seguido de la red de destino junto su máscara y, por último, indicamos la puerta de enlace para llegar a la red deseada.

5.2. Configuración de Firewalls

Para que la comunicación entre redes pueda funcionar en los diferentes firewalls como primer paso debemos configurar las diferentes IP en cada una de las diferentes patas del firewall. La pata RED que es la que sale a internet la dejamos en DHCP para que adquiera su IP de manera automática. La pata GREEN la configuramos de modo que sea la puerta de enlace de la red LAN, por ejemplo “192.168.1.1”. y la pata ORANGE (que actúa como DMZ) la configuramos como puerta de enlace de la red de conexión, por ejemplo, “10.0.0.1”.

Como segundo paso debemos realizar reglas de Firewall para permitir el intercambio de información entre las diferentes redes. En una regla de firewall se pueden configurar si queremos permitir o denegar determinadas IP o puertos o protocolos. En nuestro caso vamos a crear una regla en ambos firewalls que deje pasar todo el tráfico entre redes lo cuál en términos de seguridad no es lo más eficiente, pero para evitar problemas a la hora del intercambio de información en el proyecto es lo más efectivo.

Origen

☐ Dirección de origen (dirección MAC/IP o red):

☐ Firewall

Todos

☒ Redes estándar:

ORANGE (10.0.0.0/24)

☐ Ubicación

A1 - Anonymous Proxy

NAT

☐ Usar traducción de direcciones de red (NAT)

Destino

☐ Dirección de destino (dirección IP o red):

☐ Firewall

Todos

☒ Redes estándar:

Green (192.168.1.0/24)

☐ Ubicación

A1 - Anonymous Proxy

Protocolo

Todos

ACEPTAR

DESCARTAR

RECHAZAR

Ajustes adicionales

Remarcar:

Posición de la regla:

☐ Registrar regla

☐ Usar restricciones de tiempo

☐ Limite las conexiones simultáneas por dirección IP

☐ Tasa de límite de nuevas conexiones

Agregar

Volver

Reglas del Cortafuegos

#	Protocolo:	Origen	Registro	Destino	Acción
1	Todos	ORANGE	<input type="checkbox"/>	Green	<input checked="" type="checkbox"/>
2	Todos	Green	<input type="checkbox"/>	ORANGE	<input checked="" type="checkbox"/>
		Green ORANGE	Internet (Permitido) Internet (Permitido)	ORANGE (Permitido) Green (Bloqueado)	
Política: Permitido					

Se deben crear dos reglas en cada firewall permitiendo el tráfico de GREEN a ORANGE y viceversa.

Como último paso al igual que en el router, debemos crear sus respectivas rutas para el envío de paquetes.

Añadir una ruta

Dirección IP de host / Red:

Puerta de enlace (gateway):

Remarcar:

Activado: ☒

Agregar

Añadir una ruta

Dirección IP de host / Red:

Puerta de enlace (gateway):

Remarcar:

Activado: ☒

Agregar

6. Instalación y Configuración de Servicios

En esta sección del proyecto, se explicará en detalle cómo instalar y configurar diferentes servicios esenciales para la infraestructura de red del proyecto. La instalación y configuración adecuadas de estos servicios es esencial para garantizar un funcionamiento óptimo y seguro de la red. Se describirán los pasos específicos necesarios para la instalación de cada uno de estos servicios en los sistemas operativos correspondientes.

6.1. DHCP

DHCP es un protocolo de red que permite a los dispositivos obtener automáticamente una dirección IP única y otros parámetros de configuración de red, como la máscara de subred, la puerta de enlace predeterminada y los servidores DNS. DHCP simplifica y automatiza el proceso de asignación de direcciones IP en una red, lo que ahorra tiempo y reduce los errores de configuración manual.

Para configurar este servicio en un Windows Server como primer paso debemos instalar el rol de "Servidor DHCP" dentro de la sección agregar roles y características de nuestro servidor. Dentro de las opciones de la instalación dejamos todas las opciones por defecto, Debemos asignar una IP estática a nuestro servidor para que el DHCP funcione de manera correcta.

Asistente para agregar roles y características

Seleccionar roles de servidor

SERVIDOR DE DESTINO
WIN-K5DL99EOVQO

Antes de comenzar
Tipo de instalación
Selección de servidor
Roles de servidor
Características
Servidor DHCP
Confirmación
Resultados

Seleccione uno o varios roles para instalarlos en el servidor seleccionado.

Roles	Descripción
<input type="checkbox"/> Active Directory Lightweight Directory Services	El servidor de Protocolo de configuración dinámica de host (DHCP) permite configurar, administrar y proporcionar centralmente direcciones IP temporales e información relacionada para equipos cliente.
<input type="checkbox"/> Active Directory Rights Management Services	
<input type="checkbox"/> Atestación de mantenimiento del dispositivo	
<input type="checkbox"/> Hyper-V	
<input type="checkbox"/> Servicio de protección de host	
<input type="checkbox"/> Servicios de acceso y directivas de redes	
<input checked="" type="checkbox"/> Servicios de archivos y almacenamiento (1 de 12 in	
<input type="checkbox"/> Servicios de certificados de Active Directory	
<input type="checkbox"/> Servicios de dominio de Active Directory	
<input type="checkbox"/> Servicios de Escritorio remoto	
<input type="checkbox"/> Servicios de federación de Active Directory	
<input type="checkbox"/> Servicios de implementación de Windows	
<input type="checkbox"/> Servicios de impresión y documentos	
<input type="checkbox"/> Servidor de fax	
<input checked="" type="checkbox"/> Servidor DHCP	
<input type="checkbox"/> Servidor DNS	
<input type="checkbox"/> Servidor web (IIS)	
<input type="checkbox"/> Volume Activation Services	
<input type="checkbox"/> Windows Server Update Services	

< Anterior **Siguiente >** Instalar Cancelar

Una vez instalado debemos crear un ámbito IPv4 para asignar las diferentes opciones de red. Al ámbito lo llamaremos “clientes” y asignará las IP a los equipos clientes.

Asistente para ámbito nuevo

Nombre de ámbito

Debe escribir un nombre identificativo para el ámbito. También puede proporcionar una descripción.



Escriba un nombre y una descripción para este ámbito. Esta información le ayuda a identificar rápidamente cómo se usa el ámbito y su red.

Nombre:

Descripción:

< Atrás

Siguiente >

Cancelar

Después debemos asignar cuales son las IP que se asignarán en el ámbito así como la máscara de red.

Asistente para ámbito nuevo

Intervalo de direcciones IP

Para definir el intervalo de direcciones del ámbito debe identificar un conjunto de direcciones IP consecutivas.

Opciones de configuración del servidor DHCP

Escriba el intervalo de direcciones que distribuye el ámbito.

Dirección IP inicial: 192 . 168 . 1 . 80

Dirección IP final: 192 . 168 . 1 . 100

Opciones de configuración que se propagan al cliente DHCP

Longitud: 24

Máscara de subred: 255 . 255 . 255 . 0

< Atrás **Siguiente >** Cancelar

Ahora asignamos la dirección del enrutador que será que se proporcione a los equipos.

Asistente para ámbito nuevo

Enrutador (puerta de enlace predeterminada)

Puede especificar los enrutadores, o puertas de enlace predeterminadas, que se distribuirán en el ámbito.

Para agregar una dirección IP para un enrutador usado por clientes, escriba la dirección.

Dirección IP:

1 . . .

Agregar

192.168.1.1

Quitar

Arriba

Abajo

< Atrás **Siguiente >** Cancelar

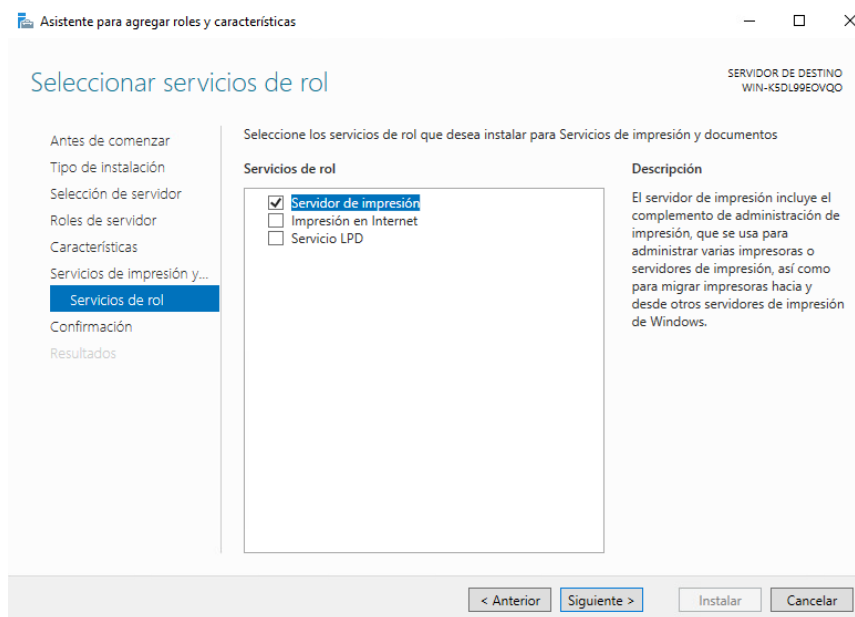
Por último, asignamos los servidores DNS que se usarán.

006 Servidores DNS	Estándar	192.168.1.3, 8.8.8.8, 8.8.4.4
--------------------	----------	-------------------------------

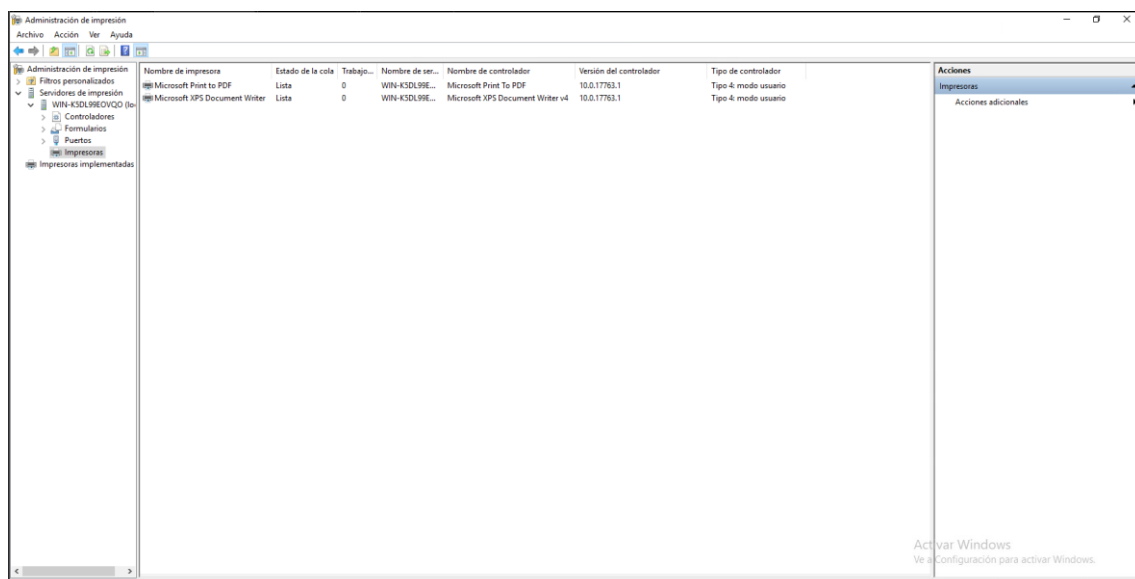
6.2. Servidor Impresión

Un servidor de impresión es un servicio que actúa como intermediario entre las impresoras y las estaciones de trabajo en una red. Su función es gestionar y controlar la impresión de documentos enviados desde las estaciones de trabajo a las impresoras, permitiendo una administración centralizada de los recursos de impresión en la red. Al utilizar un servidor de impresión, se pueden compartir una o varias impresoras en la red, lo que mejora la eficiencia y reduce los costos de impresión.

Para instalar este servicio dentro de un servidor Windows debemos instalar el rol de “Servicios de impresión y documentos” y dentro de este el servicio de “Servidor de impresión”.



Una vez instalado podemos acceder al panel de administración de impresión donde podemos ver las diferentes impresoras instaladas y agregar nuevas así como controladores o puertos.

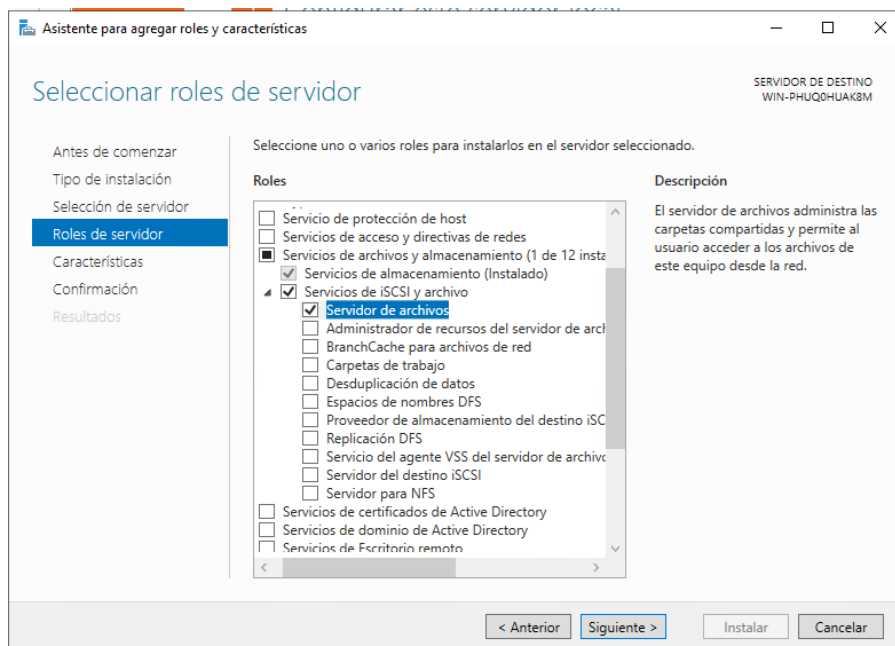


6.3. Servidor de archivos

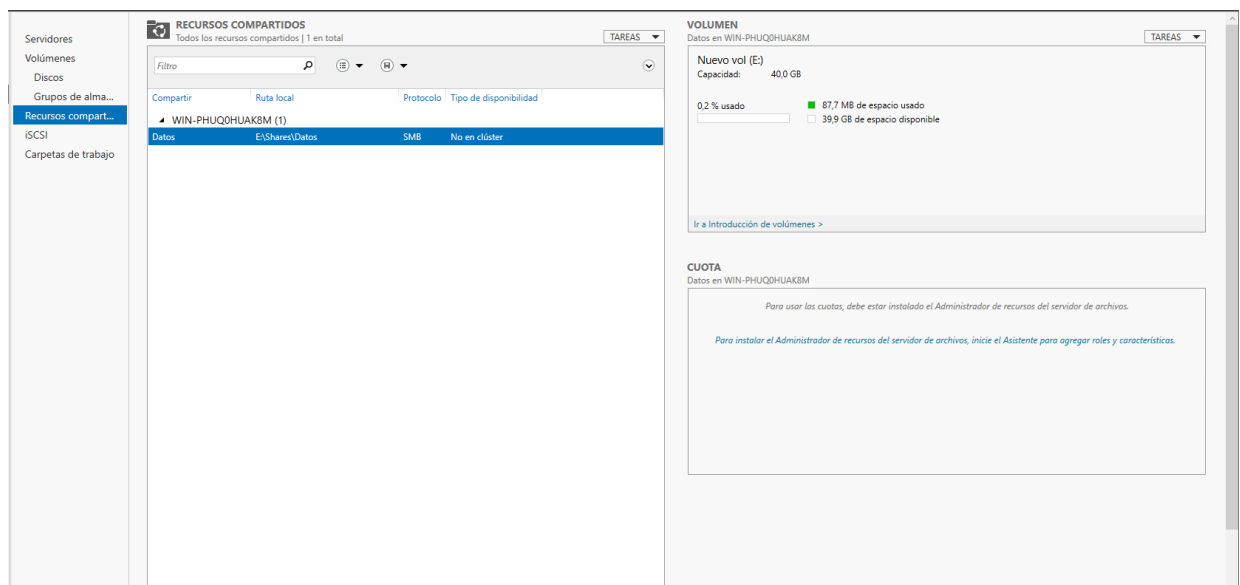
Un servidor de archivos es un dispositivo o programa que proporciona almacenamiento centralizado y compartido para los archivos y datos en una red. Su función es permitir que los usuarios accedan y compartan archivos y carpetas de manera eficiente y segura. Para proporcionar un sistema de control de acceso y permisos para garantizar la privacidad y seguridad de los datos. Al utilizar un servidor de archivos, se pueden centralizar los datos en un solo lugar, y facilitar su administración y backup.

Dentro de nuestro servidor de archivos vamos a configurar una carpeta compartida en un disco secundario llamada "DATOS" donde se podrá almacenar la información del sistema.

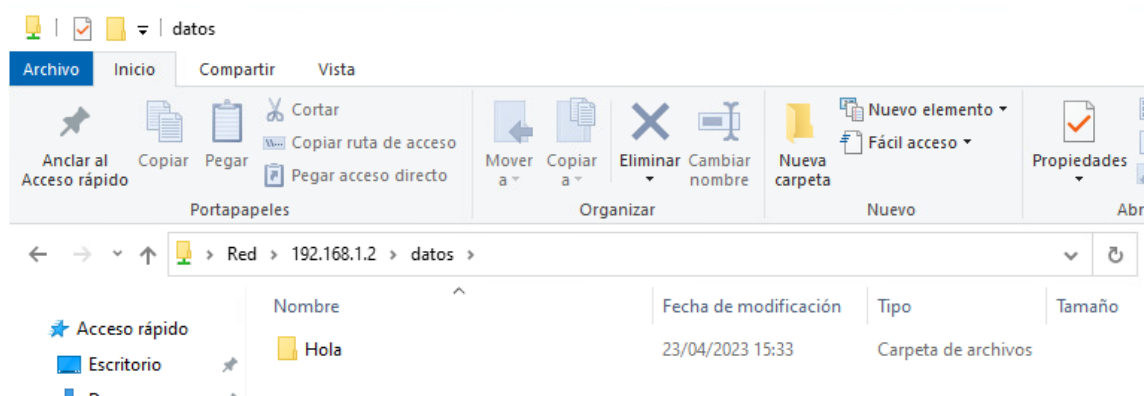
Para instalarlo dentro del servidor Windows debemos instalar el rol de "Servicios iSCSI y archivos" y dentro de este "Servidor de archivos"



Una vez instalado dentro del apartado recursos compartidos debemos generar la carpeta compartida con sus diferentes permisos.



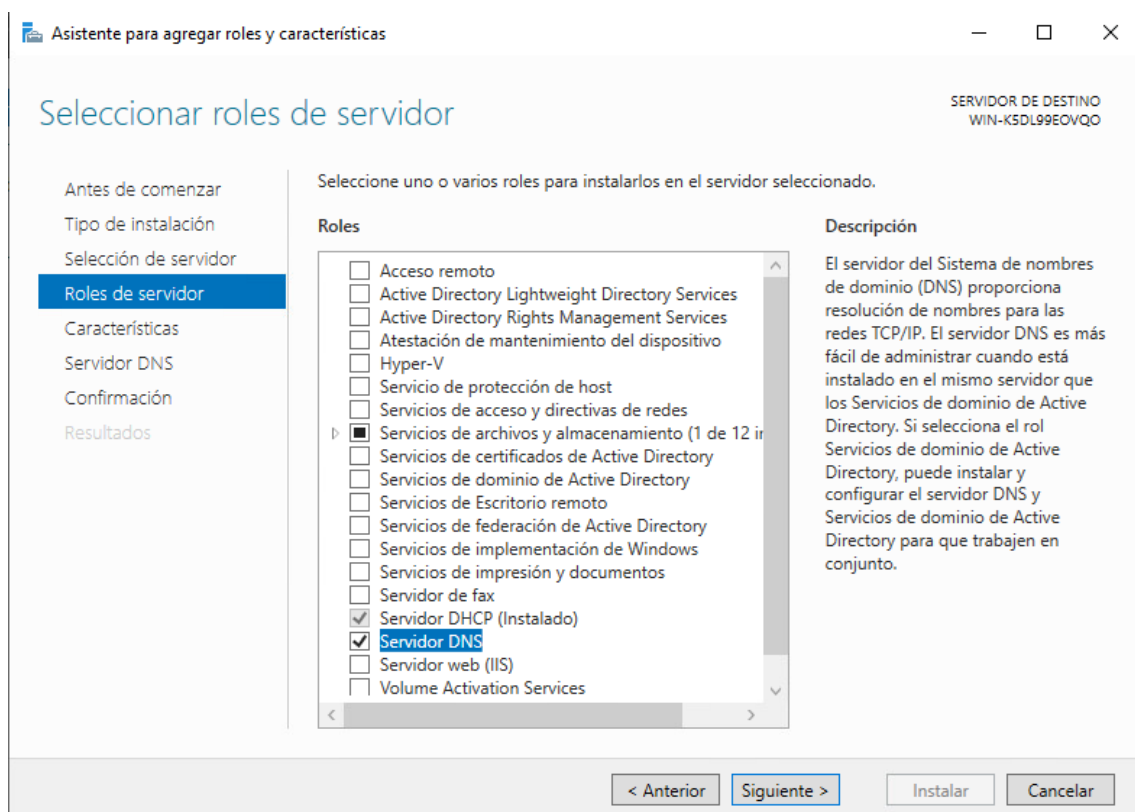
Una vez creado podemos acceder desde un equipo cliente usando el un usuario y contraseña que tenga acceso.



6.4. DNS

DNS es un sistema de nomenclatura jerárquico y distribuido que se utiliza para traducir nombres de dominio en direcciones IP y viceversa. Su función principal es permitir que los usuarios accedan a los recursos de red utilizando nombres de dominio fáciles de recordar en lugar de direcciones IP numéricas. Cuando se introduce una URL en el navegador web, el sistema DNS resuelve el nombre de dominio en la dirección IP correspondiente, lo que permite establecer la conexión con el servidor web. DNS también facilita la gestión de los nombres de dominio y la administración de los servidores DNS en una red.

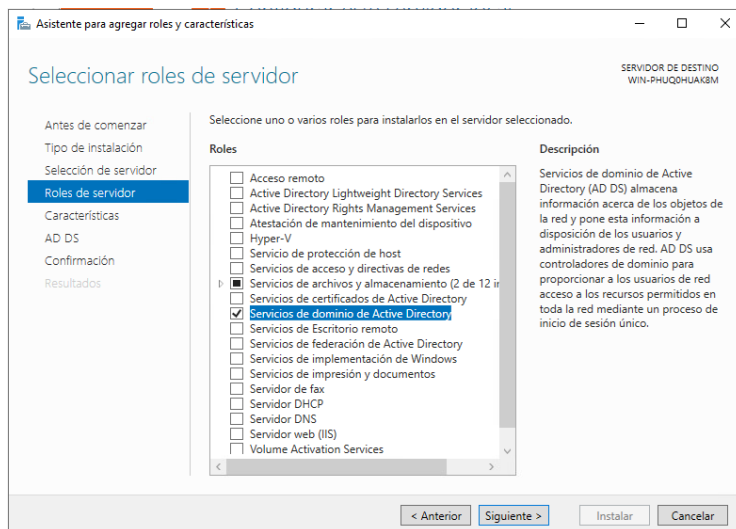
En este caso únicamente debemos instalar el rol de “Servidor DNS” y no tenemos que configurar nada ya que las zonas se crearán automáticamente al promocionar este servidor a controlador de dominio en la instalación de AD.



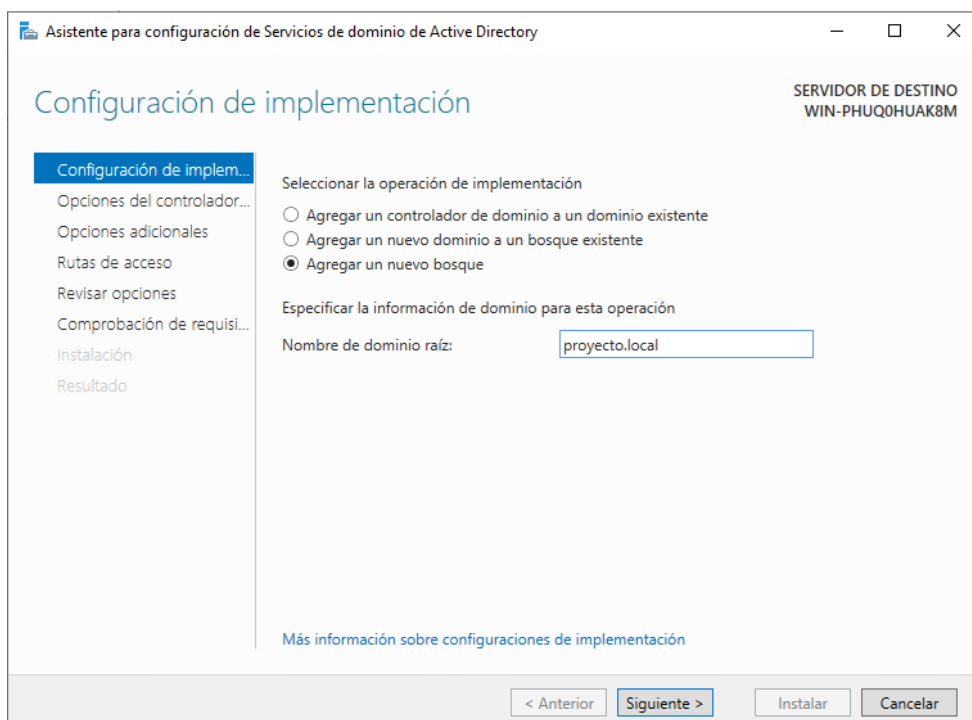
6.5. AD

AD es un servicio de directorio desarrollado por Microsoft que se utiliza para almacenar y administrar información de seguridad y recursos de red en una red de computadoras basada en Windows. Su función principal es autenticar y autorizar el acceso de los usuarios a los recursos de red, como archivos compartidos, impresoras, aplicaciones y otros servicios. AD permite la gestión centralizada de los usuarios, grupos, permisos y políticas de seguridad en una red, lo que facilita la administración y protección de los recursos de la red.

Para instalarlo en un servidor Windows debemos instalar el rol de “Servicios de dominio de Active Directory” y promocionar este servidor a controlador de dominio lo que creará las zonas DNS necesarias para que el dominio funcione con normalidad.



Una vez instalado debemos promocionar el servidor a controlador de dominio. Como primer paso debemos seleccionar la opción de agregar un nuevo bosque y proporcionar el nombre de dominio que queremos usar



Después debemos asignar el nivel funcional del dominio y la contraseña de recuperación.

The screenshot shows the 'Opciones del controlador de dominio' (Domain Controller Options) window. The title bar reads 'Asistente para configuración de Servicios de dominio de Active Directory'. The left sidebar contains a list of steps: 'Configuración de implem...', 'Opciones del controlador...' (highlighted), 'Opciones de DNS', 'Opciones adicionales', 'Rutas de acceso', 'Revisar opciones', 'Comprobación de requisi...', 'Instalación', and 'Resultado'. The main area is titled 'Opciones del controlador de dominio' and 'SERVIDOR DE DESTINO WIN-PHUQ0HUAK8M'. It includes sections for 'Seleccionar nivel funcional del nuevo bosque y dominio raíz' with dropdowns for 'Nivel funcional del bosque' and 'Nivel funcional del dominio' both set to 'Windows Server 2016'. Below is 'Especificar capacidades del controlador de dominio' with checkboxes for 'Servidor de Sistema de nombres de dominio (DNS)', 'Catálogo global (GC)', and 'Controlador de dominio de solo lectura (RODC)'. The 'Contraseña de modo de restauración de servicios de directorio (DSRM)' section has fields for 'Contraseña' and 'Confirmar contraseña'. A link 'Más información sobre opciones del controlador de dominio' is at the bottom. Navigation buttons at the bottom are '< Anterior', 'Siguiente >', 'Instalar', and 'Cancelar'.

Por último, asignamos las rutas de la información del dominio y instalamos el controlador de dominio.

The screenshot shows the 'Rutas de acceso' (Access Paths) window. The title bar reads 'Asistente para configuración de Servicios de dominio de Active Directory'. The left sidebar contains a list of steps: 'Configuración de implem...', 'Opciones del controlador...', 'Opciones de DNS', 'Opciones adicionales', 'Rutas de acceso' (highlighted), 'Revisar opciones', 'Comprobación de requisi...', 'Instalación', and 'Resultado'. The main area is titled 'Rutas de acceso' and 'SERVIDOR DE DESTINO WIN-PHUQ0HUAK8M'. It includes a section 'Especificar la ubicación de la base de datos de AD DS, archivos de registro y SYSVOL' with three rows: 'Carpeta de la base de datos:' (C:\Windows\NTDS), 'Carpeta de archivos de registro:' (C:\Windows\NTDS), and 'Carpeta SYSVOL:' (C:\Windows\SYSVOL). Each row has a text box and a browse button (...). A link 'Más información sobre rutas de acceso de Active Directory' is at the bottom. Navigation buttons at the bottom are '< Anterior', 'Siguiente >', 'Instalar', and 'Cancelar'.

Una vez instalado comprobamos que se crean las zonas DNS necesarias y podemos acceder al dominio.

	Nombre	Tipo	Datos	Marca de tiempo
DNS				
WIN-PHUQ0HUAK8M				
Zonas de búsqueda directas				
proyecto.net				
Zonas de búsqueda inversas				
1.168.192.in-addr.arpa				
Puntos de confianza				
Reenviadores condicionales				
	_msdcs			
	_sites			
	_tcp			
	_udp			
	DomainDnsZones			
	ForestDnsZones			
	(igual que la carpeta princip...	Inicio de autoridad (SOA)	[29] win-phuq0huak8m.proyecto.net, hostmaster.proy...	static
	(igual que la carpeta princip...	Servidor de nombres (NS)	win-phuq0huak8m.proyecto.net.	static
	(igual que la carpeta princip...	Host (A)	192.168.1.3	23/04/2023 14:00:00
	DESKTOP-MVCHO6N	Host (A)	192.168.1.81	23/04/2023 14:00:00
	win-phuq0huak8m	Host (A)	192.168.1.3	static

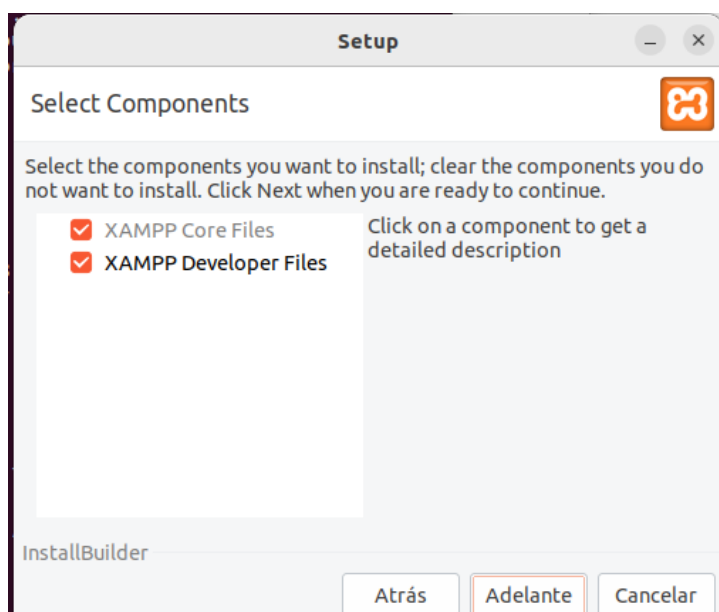
```
C:\Users\Diego>nslookup proyecto.net
Servidor: UnKnown
Address: 192.168.1.3

Nombre: proyecto.net
Addresses: 192.168.1.3
           172.16.20.3
```

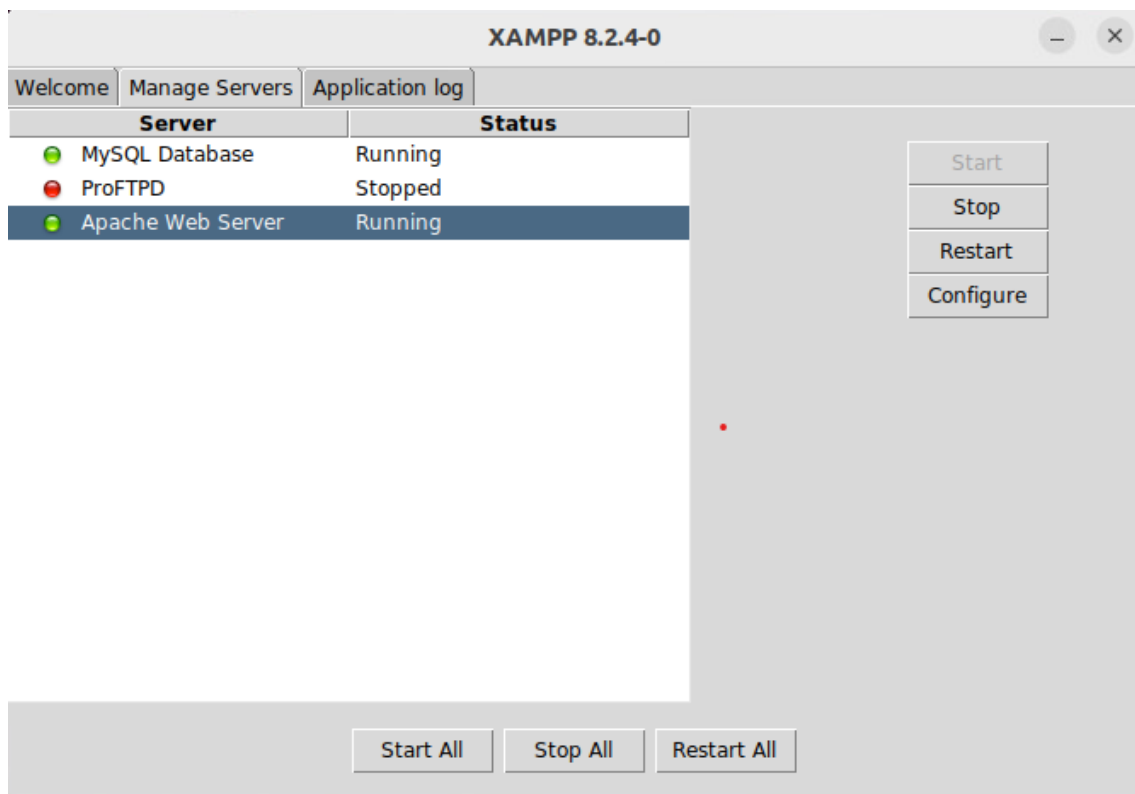
6.6. XAMPP

XAMPP es un paquete de software libre y gratuito que incluye varios componentes populares de software de servidor web, como Apache, MySQL, PHP y Perl. Su función es proporcionar un entorno de desarrollo y prueba completo para aplicaciones web y sitios dinámicos en un solo paquete fácil de instalar y configurar en diferentes sistemas operativos, como Windows, Linux y macOS. XAMPP también incluye otras herramientas útiles, como phpMyAdmin y Mercury Mail para la gestión de bases de datos y correo electrónico, respectivamente.

En este proyecto se usará para almacenar la base de datos en su MySQL y ejecutar la página web a través de el servidor Apache, se instalará en el sistema Ubuntu Desktop, para instalarlo descargamos la última versión de su página web y una vez descargado ejecutamos el archivo con el comando “./<nombre del archivo>” y nos saldrá un wizard de instalación automática.



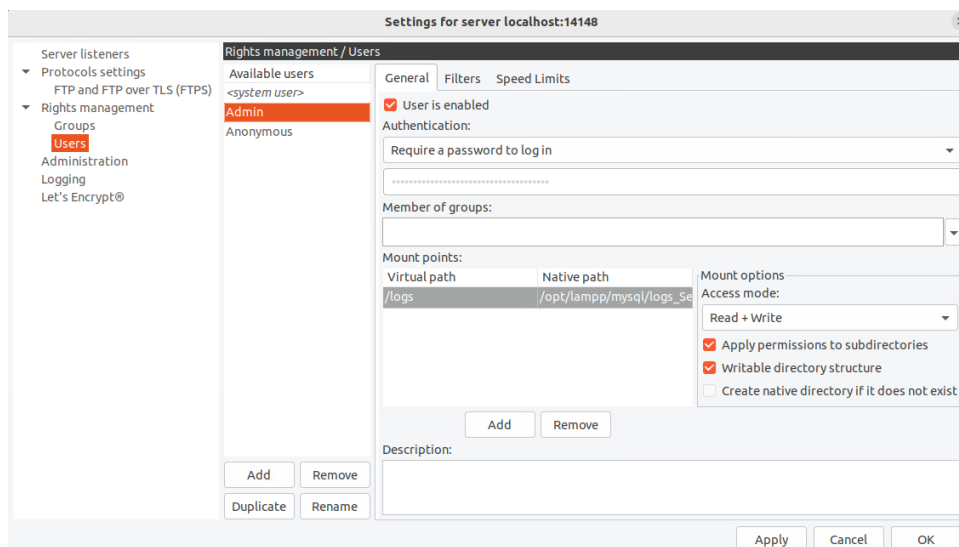
Una vez instalado accedemos al panel de control para activar los diferentes servicios.



6.7. FilleZilla Server

FileZilla Server es un servidor FTP (File Transfer Protocol) gratuito y de código abierto que se utiliza para transferir archivos entre sistemas informáticos a través de una red. Su función principal es proporcionar un servicio de transferencia de archivos seguro y eficiente a través de una conexión FTP, SFTP o FTPS.

Se va a instalar este servicio en el Ubuntu Desktop para alojar los archivos de log que se generarán a partir de las inserciones a la base de datos. Para ello descargamos el archivo de instalación desde su página web y ejecutamos con “dpkg -i <nombre del archivo>”, una vez instalado accedemos al panel de control y creamos un usuario con acceso a la carpeta de logs y le damos permisos para leer el contenido.

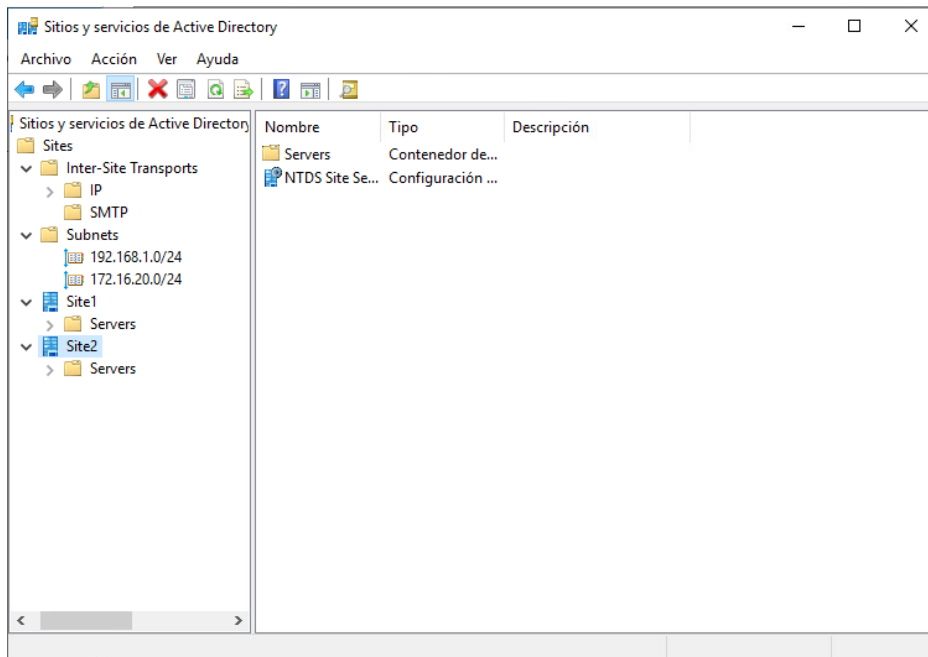


6.8. Replicación

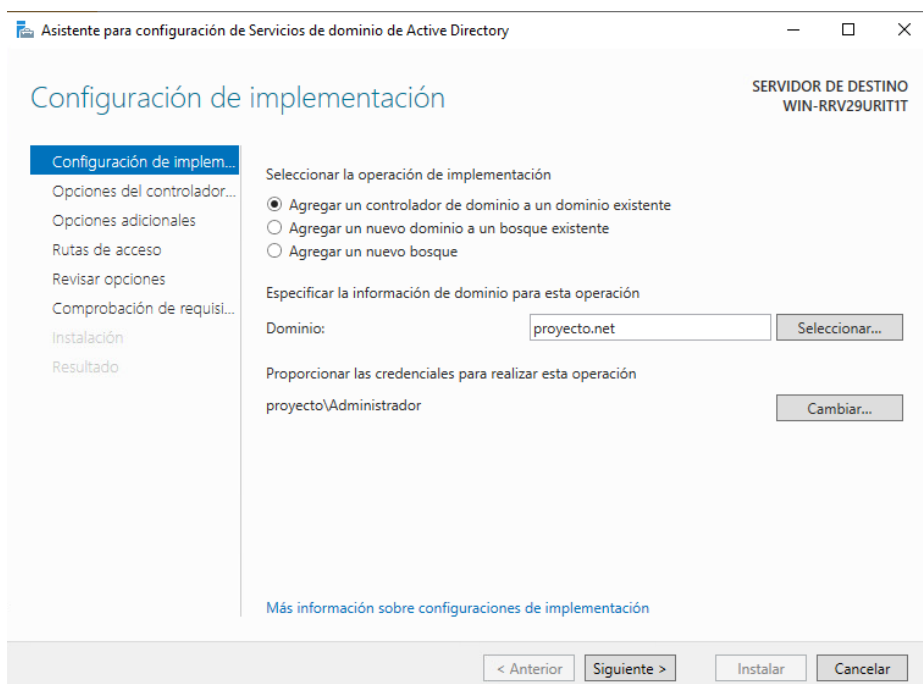
Por motivos de seguridad y backup vamos a crear replicas tanto del servidor de directorio activo como de las BBDD de los servidores. Esta replicación se realizará en los dos servidores que se encuentran en el site2 en la red 172.16.20.0/24.

6.8.1. Replicación AD

Para la realización del dominio AD lo primero que debemos hacer es abrir en el servidor principal, abrir el panel de “Sitios y servicios de Active Directory” y generar un segundo sitio para el nuevo controlador de dominio, ya que se encuentra en una nueva red, así como una nueva subred.



Ahora debemos instalar el rol de AD en el nuevo servidor y promocionar este a controlador de dominio indicando que se quiere agregar el controlador a un dominio existente y proporcionando las credenciales de administrador del dominio.



Después seleccionamos el sitio que queremos asignar y dejamos el resto de opciones por defecto:

Asistente para configuración de Servicios de dominio de Active Directory

Opciones del controlador de dominio

SERVIDOR DE DESTINO
AD2.proyecto.net

Configuración de implem...
Opciones del controlador...
Opciones de DNS
Opciones adicionales
Rutas de acceso
Revisar opciones
Comprobación de requisi...
Instalación
Resultado

Especificar capacidades del controlador de dominio e información del sitio

☒ Servidor de Sistema de nombres de dominio (DNS)
☒ Catálogo global (GC)
☐ Controlador de dominio de solo lectura (RODC)

Nombre del sitio: Site2

Escribir contraseña de modo de restauración de servicios de directorio (DSRM)

Contraseña:
Confirmar contraseña:

Más información sobre opciones del controlador de dominio

< Anterior **Siguiente >** Instalar Cancelar

Por último, comprobamos que se han generado los conectores necesarios para la replicación.

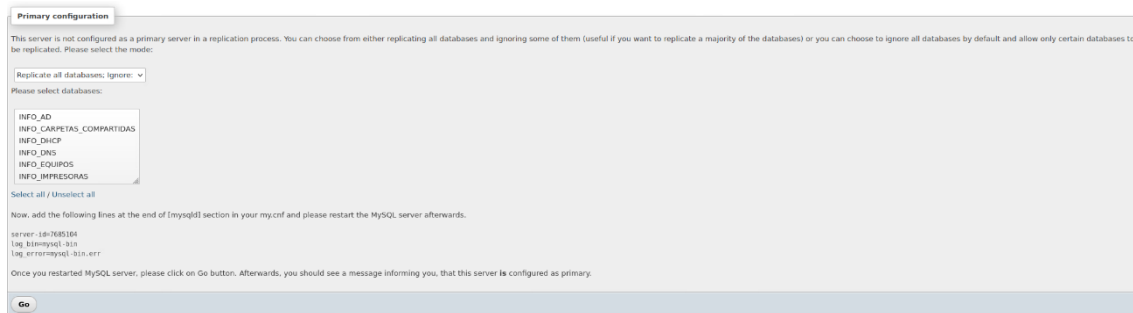
Sitios y servicios de Active Directory [WIN-PHUQ0HUAK8M.p	Nombre	Desde el servid...	Desde el sitio	Tipo	Descripción
Sitios	<generado automáticamente>	AD2	Site2	Conexión	
Inter-Site Transports					
Subnets					
Site1					
Servers					
WIN-PHUQ0HUAK8M					
NTDS Settings					
Site2					
Servers					
AD2					
NTDS Settings					

Sitios y servicios de Active Directory [AD2.proyecto.net]	Nombre	Desde el servid...	Desde el sitio	Tipo	Descripción
Sitios	<generado automáticamente>	WIN-PHUQ0H...	Site1	Conexión	
Inter-Site Transports					
Subnets					
Site1					
Servers					
WIN-PHUQ0HUAK8M					
NTDS Settings					
Site2					
Servers					
AD2					
NTDS Settings					

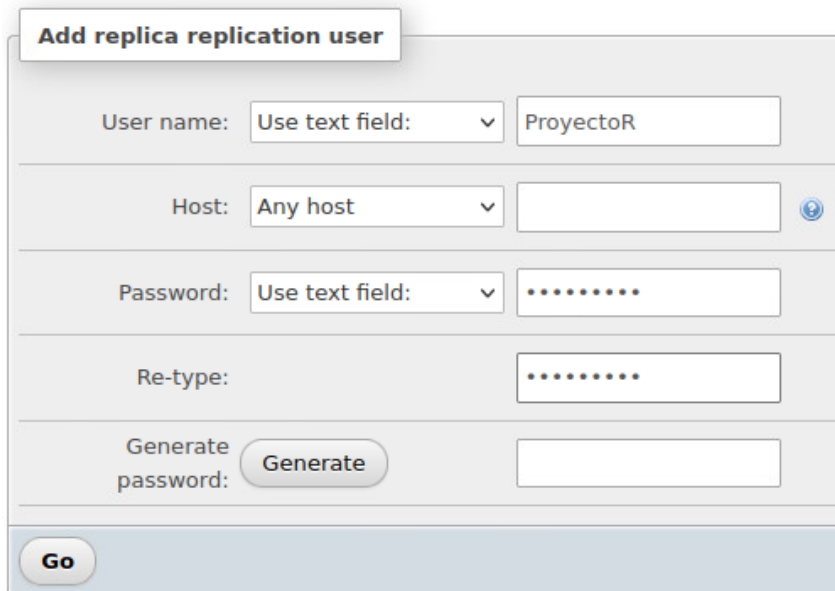
Una vez Hecho esto, nuestro dominio está replicado en ambos servidores y es accesible por cada uno de ellos independientemente, esto quiere decir que si un servidor se apagase por cualquier motivo siempre se podrá acceder al dominio a través del otro.

6.8.2. Replicación MySQL

Para replicar MySQL a través de phpMyAdmin debemos activar la replicación en la pantalla de “Replication”, nosotros seleccionamos que se repliquen todas las bases de datos.



Una vez realizado esto debemos crear un usuario para que se conecte desde el segundo servidor para poder consultar la información a replicar.



En el servidor de replicación debemos instalar el servidor MySQL y añadir estas líneas al archivo de configuración:

```
# binlog_ignore_db = include_database_name
log_bin = /var/log/mysql/mysql-bin.log
log_bin_index = /var/log/mysql/mysql-bin.log.index
relay_log = /var/log/mysql/mysql-relay-bin
relay_log_index = /var/log/mysql/mysql-relay-bin.index
```

Por último, debemos ejecutar el siguiente comando para que se inicie la replicación de la base de datos (cambiando los campos entre comillas por la información de nuestro servidor principal):

```
mysql> STOP SLAVE;
Query OK, 0 rows affected, 2 warnings (0,00 sec)

mysql> CHANGE MASTER TO MASTER_HOST='192.168.1.4', MASTER_USER='ProyectoR', MASTER_PASSWORD='Admin123.', MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS=1137;
Query OK, 0 rows affected, 8 warnings (0,04 sec)

mysql> START SLAVE;
Query OK, 0 rows affected, 1 warning (0,02 sec)
```

7. Creación de bases de datos SQL

En este apartado del proyecto, se describirá en detalle la creación de diferentes bases de datos SQL que se utilizan para almacenar la información que se recibe de una serie de scripts ps1. Estos scripts están diseñados para extraer información sobre diferentes servicios de Windows, como el estado de los servidores, el uso de la memoria, el espacio de almacenamiento disponible y otros datos relevantes.

La creación de bases de datos SQL específicas para almacenar esta información es esencial para la gestión eficiente de los datos y para permitir un fácil acceso a la información para su posterior análisis. En este apartado se describirá el proceso completo de creación de bases de datos SQL, desde la definición de los campos de la tabla hasta su configuración.

7.1. Creación general de una BBDD

Como primer punto se va a explicar un ejemplo que sirve como plantilla de cómo se crean las diferentes BBDD, explicando cada una de las sentencias utilizadas. Todas las BBDD que se han usado en el proyecto siguen la misma idea y comandos cambiando solamente aquella información que se quiere almacenar.

En los apartados específicos para cada una de las diferentes BBDD únicamente se explicará que es lo que se quiere almacenar en cada una de sus tablas respectivamente.

Como primer paso debemos comprobar si la BBDD ya existe y, si ese es el caso, borrarla. Para ello se debe usar el comando “DROP DATABASE” para indicar que borre la BBDD seguido de “IF EXISTS” para decirle que solo la borre en el caso de que exista, y por último el nombre de la BBDD, en este caso “EJEMPLO”.

```
-- Borramos la BBDD si ya existe  
DROP DATABASE if EXISTS EJEMPLO;
```

Como información adicional se debe saber que todos los comandos SQL deben terminar con el carácter “;” a menos que se indique lo contrario como más adelante en este proyecto.

Una vez hecho esto debemos de crear la BBDD, para ello usamos el comando “CREATE DATABASE” seguido del nombre de la base de datos que queremos crear.

```
-- Creamos la base de datos  
CREATE DATABASE EJEMPLO;
```

Ahora debemos colocarnos dentro de esta base de datos para poder crear las diferentes tablas que la van a formar, para ellos se usa el comando “USE” seguido del nombre de la BBDD.

```
-- Nos colocamos sobre ella  
USE EJEMPLO;
```

Una vez dentro de la BBDD podemos comenzar a crear las diferentes tablas de la BBDD, solo se explica cómo crear una de ellas ya que todas se crean de la misma manera únicamente cambiando el número de columnas, su nombre y el tipo de dato a almacenar.

Primero usamos la sentencia “CREATE TABLE” seguido del nombre de la tabla y por último abrimos y cerramos paréntesis para dentro de estos definir las diferentes columnas separadas por comas.

```
CREATE TABLE DATOS_EJEMPLO (
  -- Definimos los campos de la tabla y sus tipos de valor
  ID int auto_increment NOT NULL,
  Dato1 VARCHAR(500) NOT NULL,
  Dato2 VARCHAR(45),
  Dato3 VARCHAR(45),
  Dato4 float,
  -- Definimos claves primarias y únicas
  PRIMARY KEY (ID),
  UNIQUE (Dato1)
);
```

Dentro de los campos podemos definir diferentes tipos de datos, los que vamos a usar son:

- **INT:** valor numérico entero
- **Float:** valor numérico con decimales

Varchar: string de texto

Auto_increment significa que cada vez que se añade una nueva línea a nuestra tabla se añade automáticamente un nuevo valor consecutivo a ese campo. Por último, definimos las claves de nuestra tabla que son aquellos valores que deben ser únicos para poder identificar cada una de las filas de nuestra tabla.

7.2. BBDD INFO_AD

La primera de las BBDD que vamos a generar es aquella encargada de recopilar información del servicio de AD del Servidor Windows. Dentro de un servidor de AD se pueden monitorizar cientos de parámetros que gestionan el servidor, pero yo he elegido solo algunos que creo que son importantes dentro de un servidor AD.

Como primera tabla tenemos la encargada de almacenar la información relativa a los usuarios de AD con las columnas:

```
CREATE TABLE Usuarios_AD (
  Nombre_Distinguido VARCHAR(500) NOT NULL,
  Nombre VARCHAR(45),
  Nombre_Principal VARCHAR(500),
  PRIMARY KEY (Nombre_Distinguido)
);
```

- **Nombre_Distinguido:** es una cadena única de nombres que identifica de manera única un objeto en una jerarquía de objetos en el directorio.
- **Nombre:** nombre de usuario asignado en el dominio
- **Nombre_Principal:** nombre de usuario seguido del dominio "<usuario>@Proyecto.net"

La siguiente tabla almacena información de los diferentes grupos del AD con:

```
CREATE TABLE Grupos_AD (
  Nombre_Distinguido VARCHAR(500) NOT NULL,
  Nombre VARCHAR(200),
  Categoria VARCHAR(45),
  Alcance VARCHAR(500),
  PRIMARY KEY (Nombre_Distinguido)
);
```

- **Nombre_Distinguido:** es una cadena única de nombres que identifica de manera única un objeto en una jerarquía de objetos en el directorio.
- **Nombre:** nombre del grupo.
- **Categoría:** categoría del grupo.
- **Alcance:** identifica su alcance en el bosque o árbol de dominios.

Por último, almacenamos información de los diferentes equipos que están dados de alta en el AD:

```
CREATE TABLE Equipos_AD (
    Nombre_Distinguido VARCHAR(500) NOT NULL,
    Nombre_DNS VARCHAR(150),
    Nombre VARCHAR(45),
    PRIMARY KEY (Nombre_Distinguido)
);
```

- **Nombre_Distinguido:** es una cadena única de nombres que identifica de manera única un objeto en una jerarquía de objetos en el directorio.
- **Nombre_DNS:** nombre del equipo seguido del dominio.
- **Nombre:** nombre del equipo.

7.3. BBDD INFO_CARPETAS_COMPARTIDAS

Seguidamente creamos la BBDD encargada de almacenar la información sobre el servicio de carpetas compartidas del Windows Server. Como primera tabla almacenamos la información relativa a las diferentes carpetas compartidas creadas en el sistema con:

```
CREATE TABLE Carpetas_Compartidas (
    Nombre VARCHAR(500) NOT NULL,
    Ruta VARCHAR(45),
    Descripción VARCHAR(500),
    PRIMARY KEY (Nombre)
);
```

- **Nombre:** nombre del recurso compartido.
- **Ruta:** ruta del recurso compartido.
- **Descripción:** descripción añadida.

Como segunda tabla almacenamos la información sobre los permisos de las diferentes carpetas compartidas guardando:

- **Nombre:** nombre del permiso.
- **Usuario:** usuarios a los que afecta el permiso.
- **Tipo_Acceso:** indica si se permite el acceso o no al permiso.
- **Permisos:** permisos aplicados.

```
CREATE TABLE Permisos_Carpetas_Compartidas (
    Nombre VARCHAR(500) NOT NULL,
    Usuario VARCHAR(45),
    Tipo_Acceso VARCHAR(45),
    Permisos VARCHAR(500)
);
```

Por últimos guardamos la información de los usuarios que están conectados a una carpeta compartida:

```
CREATE TABLE Usuarios_Carpetas_Compartidas (  
    Nombre_Usuario VARCHAR(500) NOT NULL,  
    IP_Usuario VARCHAR(45)  
);
```

- **Nombre_Usuario:** nombre del usuario conectado.
- **IP_Usuario:** IP desde la cual se realiza la conexión.

7.4. BBDD INFO_DHCP

En esta BBDD se va a almacenar la información relativa al servicio de DHCP alojado en el servidor Windows. La primera tabla será la encargada de almacenar la información general de los diferentes Ámbitos DHCP con:

```
CREATE TABLE Ambitos (  
    ID VARCHAR(500) NOT NULL,  
    NOMBRE_Ambito VARCHAR(45),  
    Estado VARCHAR(45),  
    IP_Inicio VARCHAR(500),  
    IP_Final VARCHAR(500),  
    Tiempo_Concesión VARCHAR(45),  
    PRIMARY KEY (ID)  
);
```

- **ID:** ID asignado al ámbito.
- **Nombre_Ambito:** nombre asignado.
- **Estado:** estado del ámbito.
- **IP_Inicio:** IP de inicio del ámbito.
- **IP_Final:** IP final del ámbito.
- **Tiempo_Concesión:** tiempo que se asigna cada IP.

Como segunda tabla vamos a guardar la información de las opciones configuradas dentro de cada ámbito como puerta de enlace, servidores DNS, etc. Se almacena:

```
CREATE TABLE Opciones_Ambitos (  
    ID_OPCION VARCHAR(500) NOT NULL,  
    ID_Ambito VARCHAR(500),  
    NOMBRE VARCHAR(45),  
    Tipo VARCHAR(45),  
    Valor VARCHAR(500),  
    PRIMARY KEY (ID_OPCION)  
);
```

- **ID_Opción:** ID asignada a cada tipo de opción.
- **ID_Ambito:** ID del ámbito asignado a esta opción.
- **Nombre:** nombre de la opción.
- **Tipo:** tipo de dato.
- **Valor:** valor de la opción.

Ahora vamos a guardar aquellas IP que están reservadas en los diferentes ámbitos con:

```
CREATE TABLE Reservas_Ambitos (  
    IP VARCHAR(500) NOT NULL,  
    ID_Ambito VARCHAR(500),  
    ID_Cliente VARCHAR(45),  
    Nombre VARCHAR(45),  
    Tipo VARCHAR(500),  
    Descripcion VARCHAR(500),  
    PRIMARY KEY (IP)  
);
```

- **IP:** IP reservada.
- **ID_Ambito:** ID del ámbito que reserva la IP.
- **ID_Cliente:** MAC del cliente.
- **Nombre:** nombre del equipo.
- **Tipo:** tipo de reserva.
- **Descripción:** descripción añadida.

Ahora guardaremos las estadísticas de uso de ámbito:

```
CREATE TABLE Estadísticas_Ambitos (  
    ID_Ambito VARCHAR(500),  
    IP_Libres VARCHAR(45),  
    IP_Uso VARCHAR(45),  
    Reservadas VARCHAR(45)  
);
```

- **ID_Ambito:** ID del ámbito.
- **IP_Libres:** número de IPs libres.
- **IP_Uso:** número de IPs en uso.
- **Reservadas:** número de IPs reservadas.

Por último, se quiere almacenar información sobre las IPs que han sido asignadas por el servidor.

```
CREATE TABLE IP_Asignadas (  
    ID_Ambito VARCHAR(45),  
    IP_Cliente VARCHAR(45),  
    ID_Cliente VARCHAR(45),  
    Nombre VARCHAR(45),  
    Estado VARCHAR(45),  
    PRIMARY KEY (IP_Cliente)  
);
```

- **ID_Ambito:** ID del ámbito que asigna la IP.
- **IP_Cliente:** IP del cliente.
- **ID_Cliente:** MAC del cliente.
- **Nombre:** nombre del equipo.
- **Estado:** estado de la concesión.

7.5. BBDD INFO_DNS

En esta BBDD se quiere almacenar la información del servidor DNS guardando información de las diferentes zonas y registros de las mismas. Primero se guarda lo siguiente de las zonas DNS:

```
CREATE TABLE Zonas_DNS (  
  Nombre_Zona VARCHAR(45),  
  Tipo VARCHAR(45),  
  Firmada VARCHAR(45),  
  PRIMARY KEY (Nombre_Zona)  
);
```

- **Nombre_Zona:** nombre de la zona DNS.
- **Tipo:** tipo de zona.
- **Firmada:** indica si la zona está firmada o no.

Como segundo paso dentro de estas zonas se guarda la siguiente información de sus registros:

```
CREATE TABLE Registros_Zonas_DNS (  
  Nombre_Host VARCHAR(500),  
  Tipo VARCHAR(45),  
  Fecha VARCHAR(45),  
  TTL VARCHAR(45)  
);
```

- **Nombre_Host:** nombre del host.
- **Tipo:** tipo de registro.
- **Fecha:** fecha de creación.
- **TTL:** tipo de vida del registro.

7.6. BBDD INFO_EQUIPOS

En esta BBDD vamos a guardar la información relativa a los equipos servidores de nuestro site principal. En la primera tabla se guardará los datos de las 10 primeras IPs de la red para saber si están encendidas y que sistema es, se almacena:

```
CREATE TABLE Equipos_Encendidos (  
  IP VARCHAR(20) NOT NULL,  
  NOMBRE_Equipo VARCHAR(45),  
  SO VARCHAR(500),  
  Encendido BOOLEAN NOT NULL,  
  PRIMARY KEY (IP)  
);
```

- **IP:** IP del equipo.
- **Nombre_Equipo:** nombre del equipo.
- **SO:** sistema operativo del equipo.
- **Encendido:** indica si el equipo está encendido o no.

En la segunda de las tablas se almacena la información de los dos servidores principales, guardando:

```
CREATE TABLE Datos_Equipos (
    IP VARCHAR(20) NOT NULL,
    Capacidad_Disco VARCHAR(20),
    Espacio_Usado_Disco VARCHAR(20),
    Porcentaje_Uso_CPU VARCHAR(20),
    RAM_Total VARCHAR(20),
    RAM_Uso VARCHAR(20),
    PRIMARY KEY (IP)
);
```

- **IP:** IP del equipo.
- **Capacidad_Disco:** capacidad total del disco principal.
- **Espacio_Usado_Disco:** espacio usado del disco principal.
- **Porcentaje_Uso_CPU:** porcentaje de uso de la CPU.
- **RAM_Total:** RAM total del equipo.
- **RAM_Uso:** RAM en uso del equipo.

7.7. BBDD INFO_IMPRESORAS

En está BBDD se va a almacenar la información de las diferentes impresoras dentro del servidor de impresión de Windows almacenando toda la información de las impresoras, drivers o puertos. Lo primero que se quiere almacenar es información sobre las impresoras como:

```
CREATE TABLE Impresoras (
    Nombre VARCHAR(500) NOT NULL,
    NOMBRE_Equipo VARCHAR(45),
    Tipo VARCHAR(45),
    Driver VARCHAR(500),
    Puerto VARCHAR(500),
    Compartida VARCHAR(45),
    Publicada VARCHAR(45),
    PRIMARY KEY (Nombre)
);
```

- **Nombre:** nombre de la impresora.
- **Nombre_Equipo:** nombre del equipo host
- **Tipo:** tipo de impresora.
- **Driver:** driver que usa.
- **Puerto:** puerto que usa.
- **Compartida:** indica si está compartida.
- **Publicada:** indica si está publicada.

La segunda de las tablas se encarga de almacenar información sobre los drivers de impresión:

```
CREATE TABLE Drivers_Impresoras (
    Nombre VARCHAR(500) NOT NULL,
    Entorno VARCHAR(45),
    Version_Driver VARCHAR(45),
    Fabricante VARCHAR(500)
);
```

- **Nombre:** nombre del driver.
- **Entorno:** entorno para el cual está desarrollado.
- **Version_Driver:** versión del driver.
- **Fabricante:** fabricante del driver.

La siguiente tabla almacena información de los puertos de impresión creados:

```
CREATE TABLE Puertos_Impresoras (
    ID int auto_increment,
    Nombre VARCHAR(500) NOT NULL,
    Nombre_Equipo VARCHAR(45),
    Descripción VARCHAR(45),
    Tipo_monitor VARCHAR(500),
    PRIMARY KEY (ID)
);
```

- **Nombre:** nombre del puerto.
- **Nombre_Equipo:** nombre del host.
- **Descripción:** descripción del puerto.
- **Tipo_monitor:** tipo de puerto.

Por último, almacenamos las diferentes colas de impresión de las impresoras:

```
CREATE TABLE Cola_Impresión (
    ID VARCHAR(500) NOT NULL,
    Nombre_Equipo VARCHAR(45),
    Nombre_Impresora VARCHAR(45),
    Nombre_Documento VARCHAR(500),
    Fecha VARCHAR(500),
    Estado VARCHAR(500),
    PRIMARY KEY (ID)
);
```

- **ID:** ID de la cola de impresión.
- **Nombre_Equipo:** nombre del cliente.
- **Nombre_Impresora:** impresora que se usa.
- **Nombre_Documento:** nombre del documento a imprimir.
- **Fecha:** fecha de envío.
- **Estado:** estado de la impresión.

8. Logs BBDD

En este apartado se describe el proceso de creación de procedimientos SQL para la creación de logs de diferentes tablas de MySQL en un sistema Ubuntu. Los logs son una herramienta valiosa para la monitorización y el seguimiento de la actividad de una base de datos. Los procedimientos SQL permiten automatizar la creación de logs para una o varias tablas de MySQL en función de diferentes criterios como fechas, eventos específicos, entre otros. En esta sección se detallará el proceso de creación de los procedimientos SQL necesarios para crear logs para diferentes tablas de MySQL en un sistema Ubuntu, lo que permitirá mejorar la gestión y la eficiencia en el seguimiento y análisis de la actividad de la base de datos. Más adelante los logs serán accesibles mediante un servidor FTP situado en el servidor.

8.1. Ruta y formato de logs

Lo primero que debemos definir es la ruta dónde vamos a almacenar nuestros archivos ya que más adelante es la ruta que deberemos asignar en nuestros procedimientos, en este caso la ruta elegida es `"/opt/lampp/mysql/logs_Servicios/<CarpetaServicio>/"`. Se ha elegido esta ruta ya que es la ruta de instalación del servidor MySQL. Es importante saber que para poder crear archivos dentro de esta carpeta debemos tener permisos sobre ella.

Lo siguiente que debemos definir es el formato que usarán nuestros archivos, en este caso el nombre de archivo será el nombre de la tabla, la cuál es el log seguido de la fecha y hora exacta donde se ha creado este archivo. El tipo de archivo escogido es `.csv` ya que es un formato fácil de leer y de fácil importación a hojas de cálculo donde poder ver toda la información de manera rápida y eficaz.

Los archivos que se generan van a ser:

- **Ruta:** `/opt/lampp/mysql/logs_Servicios/<CarpetaServicio>/`
- **Nombre archivo:** `<NombreTabla_Año_Mes_Día_Hora_Minuto_Segundo>`

8.2. Creación de procedimientos

Al igual que en la creación de las diferentes BBDD todos los procedimientos de log que vamos a usar se han realizado siguiendo el mismo patrón, por tanto, solo se explicará uno de ellos.

Como primer paso debemos comprobar si el procedimiento ya existe y si ese el caso borrarlo, para ello se debe usar el comando `"DROP PROCEDURE"` para indicar que borre el procedimiento seguido de `"IF EXISTS"` para decirle que solo lo borre en el caso de que exista, y por último el nombre del procedimiento.

```
DROP PROCEDURE IF EXISTS INFO_AD.Log_AD;
```

Ahora debemos cambiar el delimitador para indicar cuando termina el procedimiento.

```
-- Cambiamos el delimitador por "$$"
DELIMITER $$
```

Una vez hecho esto debemos de crear el procedimiento, para ello usamos el comando `"CREATE PROCEDURE"` seguido del nombre del procedimiento que queremos crear.

```
-- Creamos un procedimiento
CREATE PROCEDURE INFO_AD.Log_AD()
```

Para iniciar el procedimiento utilizamos el comando "BEGIN" y para finalizar usamos el comando "END" seguido del delimitador que hemos definido, en este caso "\$\$". Al finalizarlo cambiamos de nuevo al delimitador por defecto.

```
-- Iniciamos el procedimiento  
BEGIN
```

```
-- Finalizamos el procedimiento  
END$$  
  
delimiter ;
```

Dentro del procedimiento debemos definir la ruta del archivo de log, lo primero que debemos hacer es declarar la variable donde se va a almacenar.

```
-- Declaramos las variables que necesitamos  
Declare Ruta varchar(100);
```

Ahora debemos crear la sentencia que almacena esa ruta y guarda la información de la tabla en un archivo que creamos.

```
SET @RUTA = concat("SELECT * FROM Usuarios_AD INTO OUTFILE '/opt/lampp/mysql/logs_Servicios/AD/Log_Usuarios_AD",  
DATE_FORMAT(NOW(), '%Y_%m_%d_%H_%i_%s'), '.csv', "' FIELDS TERMINATED BY ','");  
PREPARE statement FROM @RUTA;  
EXECUTE statement;  
TRUNCATE TABLE Usuarios_AD;
```

La variable "@RUTA" se usa para concatenar la sentencia SQL que exportará los datos a un archivo CSV, con el nombre de archivo formateado con la fecha y hora actuales. Luego, se prepara una sentencia SQL dinámica utilizando la variable "@RUTA" y se ejecuta para exportar los datos de la tabla a un archivo CSV en el directorio especificado. También se indica que los datos se separarán por comas.

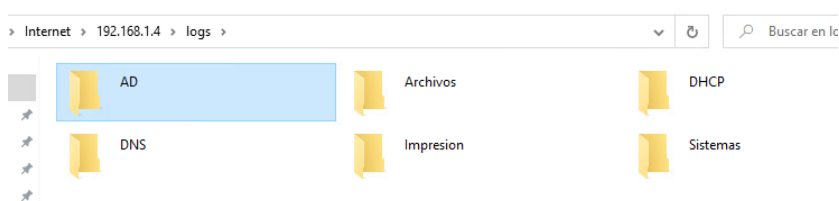
Finalmente, la tabla "Usuarios_AD" se vacía utilizando el comando TRUNCATE, lo que significa que se eliminan todos los registros de la tabla, dejándola lista para recibir nuevos datos.

Generamos un procedimiento por cada una de las BBDD y una sentencia como la anterior para cada una de las diferentes tablas dentro de la BBDD.

Los diferentes procedimientos serán lanzados desde sus respectivos scripts de powershell

8.3. Acceso a logs

Para acceder a estos archivos tenemos dos opciones. La primera es acceder directamente a la ruta donde se encuentran los archivos, el inconveniente de esta opción es que es necesario acceder al equipo directamente. La segunda opción es acceder a través del servidor FTP que se encuentra en el equipo, Debemos utilizar el usuario y contraseña creados.



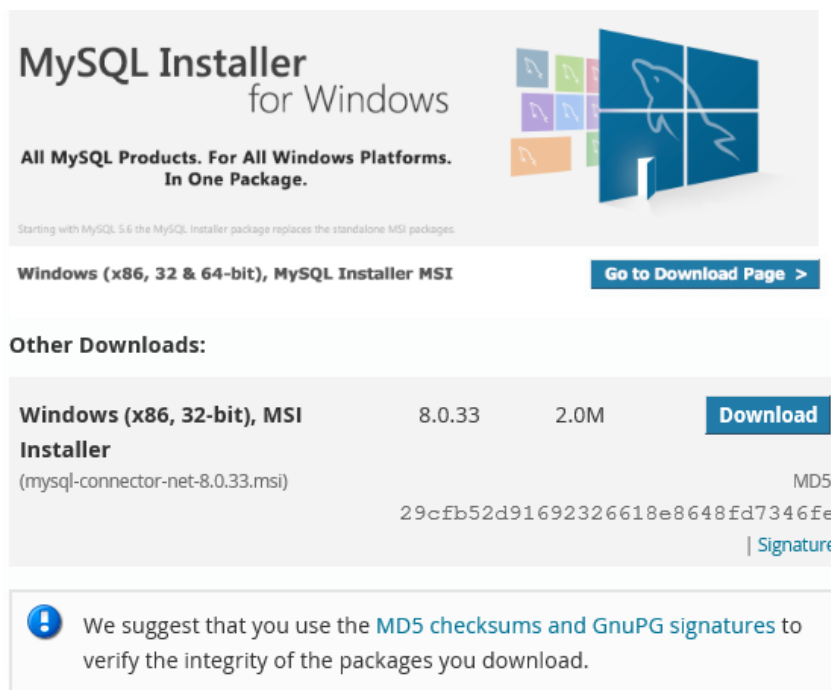
9. Creación de Scripts Powershell

En este apartado del proyecto, se describirá el proceso de creación de una serie de scripts de PowerShell que se utilizarán para extraer información de diferentes servicios de Windows. Una vez que se extrae la información, se envía a una base de datos MySQL donde se almacena y se puede analizar más adelante. El objetivo de esta sección es demostrar cómo se puede automatizar la recopilación de datos en un entorno de Windows y enviarlos a una base de datos para su posterior análisis.

Se realiza un script por cada uno de los diferentes servicios que se quieren analizar. Todos tienen el mismo formato por lo que se explicará aquello que tienen en común y luego se explicarán los diferentes cmdlets usados para la captación de información.

9.1. SQL dentro de Powershell

Para poder utilizar el módulo de MySQL dentro de PowerShell lo que debemos hacer es instalar el conector .NET de MySQL dentro de las diferentes máquinas que vayan a hacer uso de los scripts. El conector es accesible desde la página de descargas de MySQL:



MySQL Installer for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

Go to Download Page >

Other Downloads:

Windows (x86, 32-bit), MSI Installer (mysql-connector-net-8.0.33.msi)	8.0.33	2.0M	Download
--	--------	------	----------

MD5:
29cfb52d91692326618e8648fd7346fe
| Signature

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

9.2. Función de conexión a la base de datos

Lo primero que comparten todos los scripts es la función de conexión a la base de datos, siendo esta la siguiente:

```
function Conect_database {  
    #Almacenamos la contraseña que nos da el usuario en una variable  
    $passwd = "Admin123."  
    #Cargamos el módulo de MySQL  
    [void][System.Reflection.Assembly]::LoadWithPartialName("MySQL.Data")  
    #Creamos un nuevo objeto con la conexión de MySQL  
    $global:Connection = New-Object MySQL.Data.MySqlClient.MySqlConnection  
    #Creamos el texto que nos permite conectarnos a la base de datos  
    $ConnectionString = "server=" + "192.168.1.4" + ";port=3306;uid=" + "Proyecto" + ";pwd=$passwd" + ";database=INFO_DHCP"  
    $Connection.ConnectionString = $ConnectionString  
    #Abrimos la conexión  
    $Connection.Open()  
}
```

En primer lugar, se define una variable llamada “\$passwd” que almacena la contraseña para conectarse a la base de datos. Luego, se carga el módulo “MySQL.Data” para poder interactuar con la base de datos. Después, se crea un nuevo objeto de conexión a MySQL utilizando el comando “New-Object” y se almacena en la variable global “\$Connection”. A continuación, se crea una cadena de conexión a la base de datos utilizando la dirección IP del servidor MySQL, el puerto de conexión, el usuario y la contraseña. Finalmente, se establece la conexión con la base de datos mediante la llamada al método “Open()” del objeto de conexión “\$Connection”.

9.3. Ejecución de SQL a través de Powershell

Para la ejecución de los diferentes comandos se utiliza el siguiente código:

```
$sql = New-Object MySql.Data.MySqlClient.MySqlCommand
$sql.Connection = $Connection
$sql.CommandText = 'Log_DHCP()'
$sql.ExecuteNonQuery() | Out-Null
```

Primero, se crea un nuevo objeto “MySqlCommand” y se establece la conexión a la base de datos usando la propiedad “Connection”. Luego, se establece el comando SQL que se va a ejecutar en la base de datos en la propiedad “CommandText” del objeto MySqlCommand. En este caso, el comando es 'Log_DHCP()', que puede ser una función almacenada en la base de datos. Finalmente, se llama al método “ExecuteNonQuery()” del objeto “MySqlCommand” para ejecutar el comando SQL en la base de datos, lo que actualiza o modifica los datos en la base de datos según corresponda.

El operador de redirección “Out-Null” se utiliza para descartar cualquier resultado que se genere durante la ejecución de la consulta SQL, ya que, en este caso, no se espera que se devuelva ningún resultado.

Cuando el resultado de un cmdlet devuelve más de una línea de información se utiliza un bucle foreach para almacenar la información en la base de datos línea por línea.

9.4. Cmdlets utilizados

Ahora vamos a explicar los diferentes cmdlets que hemos utilizado.

9.4.1. AD

- **Get-AdUser -Filter *:** este cmdlet se utiliza para obtener información sobre los usuarios en un dominio de Active Directory. Al utilizar el filtro “*”, se recuperarán todos los usuarios del dominio. Si se desea recuperar información específica de un usuario en particular, se puede utilizar un filtro más específico.
- **Get-AdGroup -Filter *:** este cmdlet se utiliza para obtener información sobre los grupos en un dominio de Active Directory. Al utilizar el filtro “*”, se recuperarán todos los grupos del dominio. Si se desea recuperar información específica de un grupo en particular, se puede utilizar un filtro más específico.
- **Get-AdComputer -Filter *:** este cmdlet se utiliza para obtener información sobre los equipos en un dominio de Active Directory. Al utilizar el filtro “*”, se recuperarán todos los equipos del dominio. Si se desea recuperar información específica de un equipo en particular, se puede utilizar un filtro más específico.

9.4.2. DHCP

- **Get-DHCPServerv4Scope -ComputerName 192.168.1.2:** obtiene información sobre todos los ámbitos IPv4 en un servidor DHCP específico. El parámetro -ComputerName indica el nombre o dirección IP del servidor DHCP que se desea consultar.
- **Get-DHCPServerv4Scope:** Obtiene información sobre todos los ámbitos IPv4 en el servidor DHCP local.
- **Get-DhcpServerv4OptionValue -ComputerName 192.168.1.2 -ScopeID \$Ambito.ScopeId:** obtiene los valores de las opciones de configuración del servidor DHCP para un ámbito IPv4 específico. El parámetro -ComputerName indica el nombre o dirección IP del servidor DHCP que se desea consultar, mientras que el parámetro -ScopeId especifica el identificador del ámbito IPv4.
- **Get-DhcpServerv4Reservation -ComputerName 192.168.1.2 -ScopeID \$Ambito.ScopeId:** obtiene información sobre todas las reservas de direcciones IP en un ámbito IPv4 específico. El parámetro -ComputerName indica el nombre o dirección IP del servidor DHCP que se desea consultar, mientras que el parámetro -ScopeId especifica el identificador del ámbito IPv4.
- **Get-DHCPServerv4ScopeStatistics -ComputerName 192.168.1.2:** obtiene estadísticas de un ámbito IPv4 específico en un servidor DHCP específico. El parámetro -ComputerName indica el nombre o dirección IP del servidor DHCP que se desea consultar.

9.4.3. Carpetas Compartidas

- **Get-SmbShare:** este cmdlet se utiliza para recuperar una lista de todas las carpetas compartidas de SMB en un servidor.
- **Get-SmbShareAccess -Name \$Carpeta.Name:** este cmdlet se utiliza para recuperar la lista de usuarios y grupos que tienen acceso a una carpeta compartida SMB en particular. El parámetro -Name se utiliza para especificar el nombre de la carpeta compartida a la que se desea acceder.
- **Get-SmbSession:** este cmdlet se utiliza para recuperar información sobre todas las sesiones SMB (Server Message Block) activas en un servidor.

9.4.4. DNS

- **Get-DnsServerZone -ComputerName 192.168.1.3:** este cmdlet se utiliza para obtener información sobre las zonas DNS que se encuentran en un servidor DNS específico. Al especificar el parámetro ComputerName, se puede indicar la dirección IP o el nombre de host del servidor DNS remoto del que se desea obtener información. Al ejecutar este cmdlet, se mostrará una lista de todas las zonas DNS que se encuentran en el servidor especificado, junto con su tipo y estado.
- **Get-DnsServerResourceRecord -ZoneName \$Zona.ZoneName:** este cmdlet se utiliza para obtener información sobre los registros de recursos DNS que se encuentran en una zona DNS específica. Al especificar el parámetro ZoneName, se puede indicar el nombre de la zona DNS de la que se desea obtener información. Al ejecutar este cmdlet, se mostrará una lista de todos los registros de recursos que se encuentran en la zona especificada, junto con su tipo, nombre, valor y otros detalles relevantes.

9.4.5. Servidor Impresión

- **Get-PrinterPort -ComputerName 192.168.1.2:** devuelve información sobre los puertos de impresora configurados en el equipo remoto 192.168.1.2. Este cmdlet permite

obtener información sobre los puertos en los que están conectadas las impresoras y cómo están configurados.

- **Get-PrinterDriver -ComputerName 192.168.1.2:** devuelve información sobre los controladores de impresora instalados en el equipo remoto 192.168.1.2. Este cmdlet permite obtener información sobre los controladores de impresora instalados en el equipo y sus versiones.
- **Get-Printer -ComputerName 192.168.1.2:** devuelve información sobre las impresoras instaladas en el equipo remoto 192.168.1.2. Este cmdlet permite obtener información sobre las impresoras instaladas en el equipo, como el nombre, la ubicación, el estado y los controladores utilizados.
- **Get-Printjob -ComputerName 192.168.1.2 -PrinterName \$Impresora.Name:** devuelve información sobre los trabajos de impresión pendientes en la impresora indicada. Este cmdlet permite obtener información sobre los trabajos de impresión en una impresora específica del equipo remoto 192.168.1.2, identificándola a través del nombre de impresora que se ha almacenado previamente en la variable \$Impresora.

9.4.6. Sistemas

- **Test-Connection \$PC -Count 1 -Quiet:** este cmdlet permite verificar si un equipo (\$PC) está en línea realizando un ping y devuelve un valor booleano (Verdadero/Falso) indicando si el equipo está accesible o no. El parámetro "-Count" indica el número de intentos de ping a realizar y el parámetro "-Quiet" hace que el cmdlet no genere salida por pantalla.
- **Get-WmiObject -Class Win32_LogicalDisk | Where-Object {\$_.DeviceID -eq "C:"} | Select-Object -Property DeviceID, @{Name="Capacity";Expression="{0:N2}" -f(\$_.Size/1GB)}, @{Name="FreeSpace";Expression="{0:N2}" -f(\$_.FreeSpace/1GB)}:** este cmdlet devuelve información sobre el espacio en disco disponible y utilizado en la unidad C: del equipo actual. El cmdlet utiliza el proveedor WMI para obtener información sobre los discos lógicos del sistema, filtra los resultados para mostrar sólo la unidad C: y, finalmente, formatea la salida utilizando la sintaxis @{Name="Nombre";Expression={Expresión}}. En este caso, se utilizan dos expresiones para mostrar la capacidad y el espacio libre en gigabytes con dos decimales.
- **Get-Counter '\VM Processor(Total)% Processor Time' -SampleInterval 2 | Select-Object -ExpandProperty CounterSamples | Select-Object -Property Path, @{Name='CookedValue';Expression={\$_.CookedValue.ToString("0.00")}}:** este cmdlet permite monitorizar el porcentaje de uso de CPU del equipo actual durante un intervalo de tiempo especificado por el parámetro "-SampleInterval". El cmdlet utiliza el contador de rendimiento "\VM Processor(_Total)% Processor Time" y devuelve los resultados en forma de objetos CounterSample. El cmdlet también utiliza la sintaxis @{Name="Nombre";Expression={Expresión}} para dar un nombre legible a la propiedad "CookedValue" y formatearla con dos decimales.
- **(Get-CimInstance -ClassName CIM_PhysicalMemory | Measure-Object -Property Capacity -Sum).Sum / 1GB:** este cmdlet muestra la cantidad total de memoria RAM instalada en el equipo actual. Utiliza el proveedor CIM (Common Information Model) para obtener información sobre la memoria física y utiliza el cmdlet Measure-Object para sumar la capacidad de todas las tarjetas de memoria instaladas. Finalmente, la salida se divide por 1GB para mostrar la capacidad en gigabytes.
- **Get-CimInstance -ClassName Win32_OperatingSystem | Select-Object TotalVisibleMemorySize, FreePhysicalMemory | ForEach-Object**

{{(\$TotalVisibleMemorySize - \$FreePhysicalMemory) / 1MB}}: este cmdlet devuelve información sobre el uso de la memoria RAM del equipo actual. Utiliza el proveedor CIM y la clase Win32_OperatingSystem para obtener información sobre la memoria visible total y la memoria libre, y luego utiliza una expresión para calcular la memoria utilizada restando la memoria libre de la memoria total. Finalmente, la salida se divide por 1MB para mostrar la capacidad en megabytes.

10. Automatización de Scripts PowerShell

En este apartado, se describe el proceso de automatización de los scripts de PowerShell mediante el Programador de tareas de Windows. Una vez que se han creado los diferentes scripts para realizar tareas específicas, es importante automatizar su ejecución para garantizar que se ejecuten de manera regular y consistente. El Programador de tareas de Windows proporciona una solución integrada para programar y ejecutar scripts de PowerShell en función de diferentes eventos y horarios en un sistema Windows. Esta automatización puede ayudar a mejorar la eficiencia y la productividad en la gestión de sistemas y servicios en una red de Windows. En esta sección, se detallará el proceso de configuración y programación de tareas en el Programador de tareas de Windows para automatizar la ejecución de los scripts. Se han dividido los scripts en dos tipos, aquellos que se deben ejecutar cada día ya que su información no suele cambiar rápidamente como los ámbitos creados en el servidor DHCP y aquellos que se ejecutarán cada cinco minutos ya que cambian de valor muy a menudo como la información del sistema.

Para crear una tarea debemos dirigirnos al programador de tareas y una vez allí crear una nueva tarea. Como primer paso debemos seleccionar el desencadenador de la tarea, en este caso será una marca de tiempo cada media noche, También indicamos que se ejecute cada día.

Editar desencadenador

Iniciar la tarea: Según una programación

Configuración

☐ Una vez

☒ Diariamente

☐ Semanalmente

☐ Mensualmente

Inicio: 26/04/2023 23:59:00 ☐ Sincronizar zonas

Repetir cada: 1 días

Configuración avanzada

☐ Retraso máx. (retraso aleatorio): 1 hora

☐ Repetir cada: 1 hora durante: 1 día

☐ Detener todas las tareas en ejecución al final de la duración de repetición

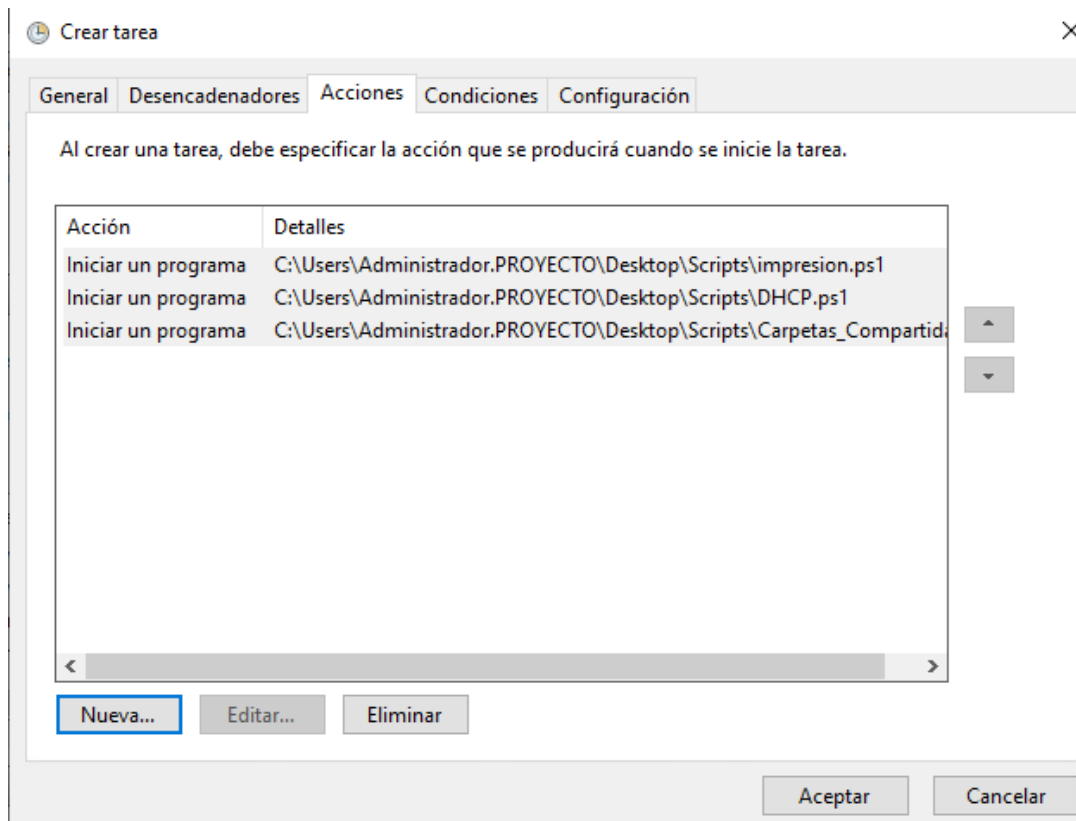
☐ Detener la tarea si se ejecuta durante más de: 3 días

☐ Expiración: 26/04/2024 15:10:12 ☐ Sincronizar zonas horaria

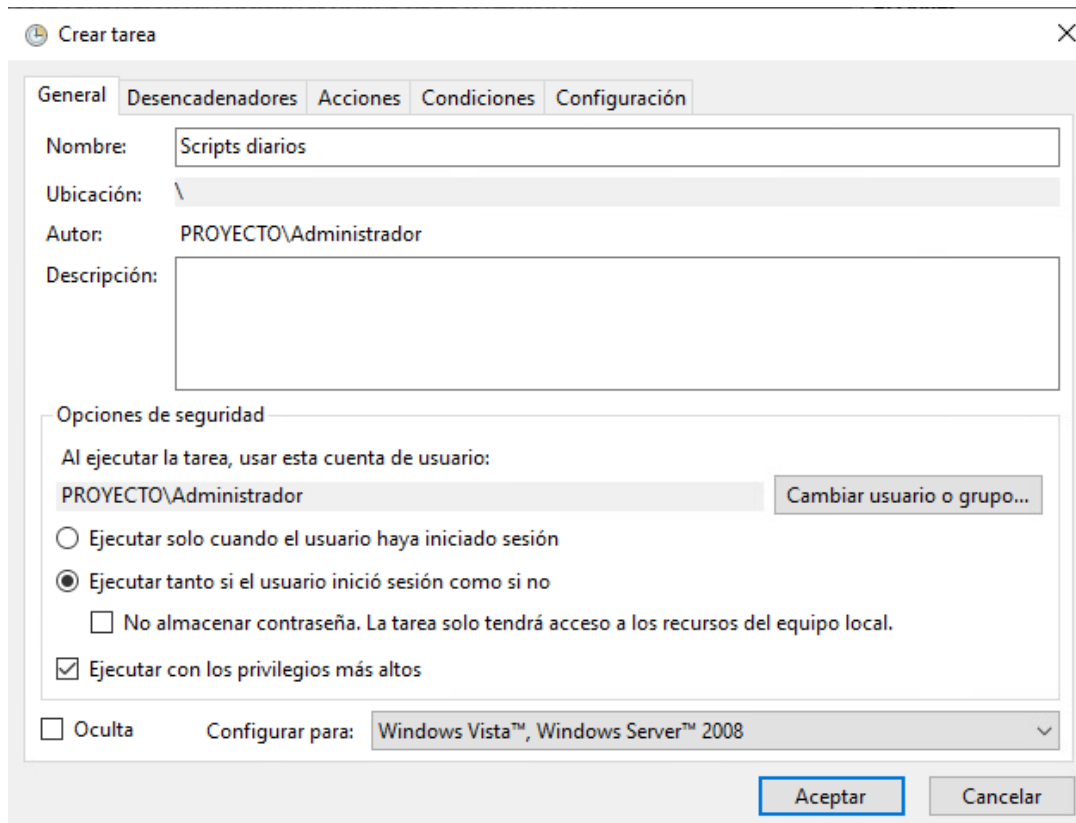
☒ Habilitado

Aceptar Cancelar

Después indicamos que va a realizar la tarea, en este caso ejecutar una serie de scripts.



Por último, asignamos un nombre a la tarea y le proporcionamos las credenciales de un usuario con permisos para realizar la acción propuesta.



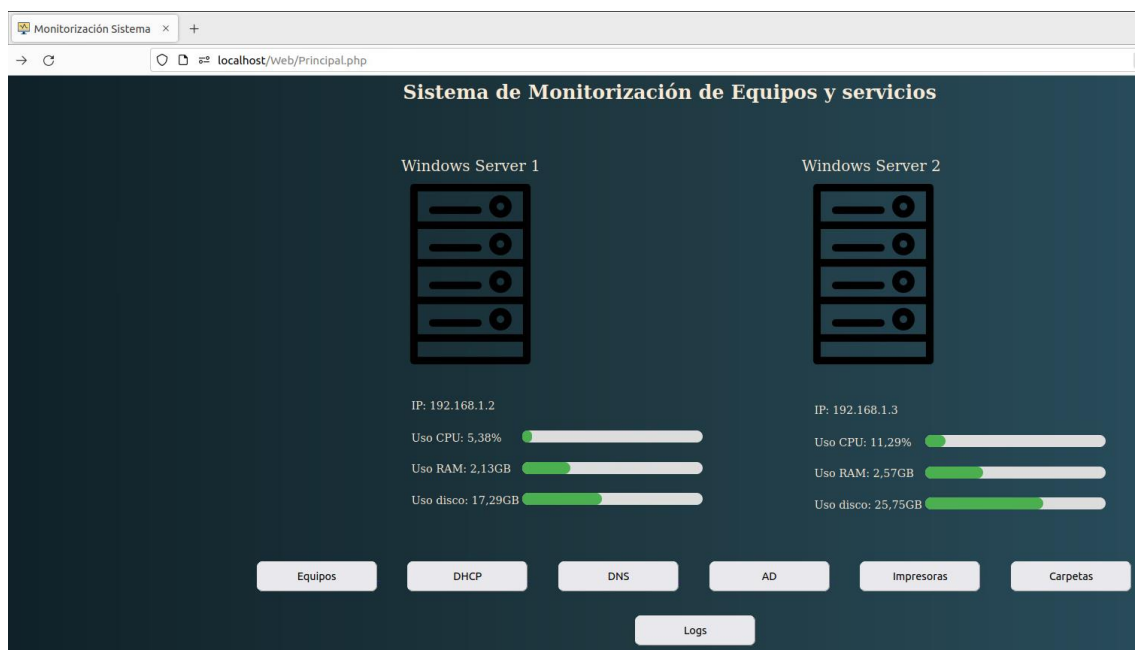
Script cada 5... Listo	A las 23:59 todos los días - Tras desencadenarse, repetir cada 5 minutos indefinidamente.	26/04/2023 23:59:00
Scripts diarios Listo	A las 23:59 todos los días	26/04/2023 23:59:00

11. Creación de la web

En este apartado se explicará el proceso de creación de una página web que permita visualizar y analizar la información obtenida de la base de datos del proyecto. Esta web se diseñará con el objetivo de facilitar la comprensión y el análisis de la información recopilada, permitiendo a los usuarios acceder a los datos de manera intuitiva y eficiente. Para ello, se utilizarán tecnologías y herramientas adecuadas para el desarrollo web.

11.1. Página principal

La página principal muestra información de los recursos de los dos servidores principales y una serie de barras para ver de manera gráfica su uso, así como una serie de botones para acceder a la información de los diferentes servicios y un último botón para acceder al servidor FTP con los diferentes logs. El resultado de la página web es el siguiente:



El código de esta página está dividido en dos archivos, uno PHP para mostrar la página y realizar las consultas a la base de datos para ver la información del sistema y un archivo CSS para dar formato a la página. Primero se explica el PHP:

```
<html>
  <title>Monitorización Sistema</title>

  <head>
    <link rel="icon" href="icono.png" type="image/png">
    <link rel="stylesheet" type="text/css" href="Estilos.css">
  </head>
```

Esta parte contiene la sección del encabezado. En ella se especifican elementos clave como el título de la página y los enlaces a hojas de estilo externas e imágenes.

```
<body>
  <?php
    // Establecer conexión
    $conn = mysqli_connect("192.168.1.4", "Proyecto", "Admin123.", "INFO_EQUIPOS");
  ?>
```

En esta sección abrimos la conexión con la base de datos.

```
<h1>Sistema de Monitorización de Equipos y servicios</h1> <br><br>

<div style="display: flex; width: 1500px; height: 30px; margin-left: 30.5%;">
  <p style="color: antiquewhite; font-size: 25px; margin-right: 435px;">Windows Server 1</p>
  <p style="color: antiquewhite; font-size: 25px;">Windows Server 2</p>
</div><br><br>

<div style="display: flex; width: 300px; height: 300px; margin-left: 29%;">
  
  
</div><br><br>
```

La primera línea muestra un encabezado principal (h1) que dice "Sistema de Monitorización de Equipos y servicios". Luego, hay dos elementos div que se utilizan para dividir y organizar el contenido de la página web.

El primer div tiene una propiedad de estilo "display: flex" que se utiliza para mostrar los elementos secundarios en una línea horizontal. Hay dos elementos secundarios, ambos de tipo párrafo (p) que contienen el nombre de los servidores ("Windows Server 1" y "Windows Server 2"). Estos párrafos están separados por un margen derecho de 435px.

El segundo div tiene una propiedad de estilo "display: flex" que también se utiliza para mostrar los elementos secundarios en una línea horizontal. Hay dos elementos secundarios, ambos de tipo imagen (img) que muestran la imagen de un servidor (Servidor.png). Estas imágenes tienen un tamaño de 300px x 300px y están separadas por un margen derecho de 370px.

```
<?php
$sql = "SELECT * FROM Datos_Equipos Where IP='192.168.1.2'";
$resultado = mysqli_query($conn, $sql);

$resultado = mysqli_fetch_assoc($resultado);

echo "<p class='Datos' style='margin-left: 715px; font-size: 18px;'> IP: " . $resultado["IP"] . "</p><br>";

$Ram_total = $resultado["RAM_Total"];
$Ram_USO = $resultado["RAM_Uso"];
$Ram_total = str_replace(',', '.', $Ram_total );
$Ram_USO = str_replace(',', '.', $Ram_USO );
$porcentaje_RAM = floatval(($Ram_USO*100)/$Ram_total);
if($porcentaje_RAM > 90){$Color_RAM = "red";}else{$Color_RAM = "#4CAF50";}

$porcentaje_CPU = $resultado["Porcentaje_Uso_CPU"];
$porcentaje_CPU = str_replace(',', '.', $porcentaje_CPU );
$porcentaje_CPU = floatval($porcentaje_CPU);
if($porcentaje_CPU > 90){$Color_CPU = "red";}else{$Color_CPU = "#4CAF50";}

$Disco_total = $resultado["Capacidad_Disco"];
$Disco_USO = $resultado["Espacio_Usado_Disco"];
$Disco_total = str_replace(',', '.', $Disco_total );
$Disco_USO = str_replace(',', '.', $Disco_USO );
$porcentaje_Disco = floatval(($Disco_USO*100)/$Disco_total);
if($porcentaje_Disco > 90){$Color_Disco = "red";}else{$Color_Disco = "#4CAF50";}
?>
```

Este código es una sección de código PHP que se utiliza para realizar una consulta a una base de datos MySQL y extraer algunos datos de ella.

Primero, se establece una conexión con la base de datos (que debe haber sido previamente configurada). Luego, se realiza una consulta SQL para seleccionar todos los datos de la tabla "Datos_Equipos" donde la dirección IP sea "192.168.1.2".

Después de ejecutar la consulta, el resultado se guarda en una variable llamada "\$resultado". Luego se utiliza la función "mysqli_fetch_assoc()" para extraer la primera fila del resultado como un array asociativo.

A continuación, se realizan algunas operaciones en los datos extraídos para calcular el porcentaje de uso de la RAM, la CPU y el disco, y se asigna un color según si el porcentaje es mayor o menor que 90%.

Por último, se imprimen los valores de la dirección IP y los porcentajes de uso de la RAM, la CPU y el disco en un formato específico en el que se especifica la etiqueta HTML de "p" con una clase "Datos" y algunos estilos CSS específicos para el tamaño y la alineación del texto.

```
<div class="barra" style="margin-left: 900px; margin-top: 7px;">
  <div class="relleno" style="width: <?php echo $porcentaje_CPU; ?>%; background-color: <?php echo $Color_CPU; ?>;"></div>
</div><br><br>

<div style="display: flex; width: 300px; height: 10px; margin-left: 715px; margin-top: -25px;">
  <?php
    echo "<p class='Datos' style='font-size: 18px;'> Uso RAM: " . $resultado["RAM_Uso"] . "GB</p>";
  >
</div>

<div class="barra" style="margin-left: 900px; margin-top: 7px;">
  <div class="relleno" style="width: <?php echo $porcentaje_RAM; ?>%; background-color: <?php echo $Color_RAM; ?>;"></div>
</div><br><br>

<div style="display: flex; width: 300px; height: 10px; margin-left: 715px; margin-top: -25px;">
  <?php
    echo "<p class='Datos' style='font-size: 18px;'> Uso disco: " . $resultado["Espacio_Usado_Disco"] . "GB";
  >
</div>

<div class="barra" style="margin-left: 900px; margin-top: 7px;">
  <div class="relleno" style="width: <?php echo $porcentaje_Disco; ?>%; background-color: <?php echo $Color_Disco; ?>;"></div>
</div><br><br>
```

Este código muestra la visualización de tres barras de progreso que representan el uso de la CPU, RAM y disco duro de un equipo con una dirección IP específica.

La primera línea de código crea una barra de progreso para el uso de la CPU. El ancho de la barra de progreso se establece dinámicamente utilizando la variable \$porcentaje_CPU, que se calcula anteriormente en el código. La propiedad de estilo background-color de la barra de progreso se establece también dinámicamente utilizando la variable \$Color_CPU.

Las siguientes líneas de código crean una pequeña sección que muestra el uso actual de RAM. El valor se muestra como un texto y se coloca en una posición específica utilizando el estilo CSS margin-left.

La siguiente sección del código crea una barra de progreso para el uso de la RAM. Al igual que la primera barra de progreso, el ancho de la barra de progreso se establece dinámicamente utilizando la variable \$porcentaje_RAM y el color de fondo utilizando la variable \$Color_RAM.

La última sección del código crea una barra de progreso para el uso del disco duro. El ancho de la barra de progreso se establece dinámicamente utilizando la variable \$porcentaje_Disco y el

color de fondo se establece dinámicamente utilizando la variable \$Color_Disco. También se muestra una sección adicional que indica el uso actual del disco duro.

Ahora esta sección PHP y los diferentes div se duplican para mostrar la información del otro servidor. Por último, cerramos la conexión y creamos los botones:

```
<?php
// Cerrar conexión
mysqli_close($conn);
?>
```

```
<a href="Equipos.php">
  <button class="boton" style="margin-left: 20%; margin-top: 2%">Equipos</button>
</a>

<a href="DHCP.php">
  <button class="boton" style="margin-left: 2%; margin-top: 2%">DHCP</button>
</a>

<a href="DNS.php">
  <button class="boton" style="margin-left: 2%; margin-top: 2%">DNS</button>
</a>

<a href="AD.php">
  <button class="boton" style="margin-left: 2%; margin-top: 2%">AD</button>
</a>

<a href="Impresoras.php">
  <button class="boton" style="margin-left: 2%; margin-top: 2%">Impresoras</button>
</a>

<a href="Carpetas.php">
  <button class="boton" style="margin-left: 2%; margin-top: 2%">Carpetas</button>
</a><br><br><br>

<a href="ftp://192.168.1.4" style="margin-left: 47.5%; margin-top: 4%">
  <button class="boton">Logs</button>
</a>
```

Este código genera una serie de botones que enlazan a diferentes páginas web. Cada botón está dentro de un enlace <a> que tiene una URL correspondiente a la página a la que lleva el botón. Los botones tienen un estilo definido en la clase "boton" que incluye propiedades de CSS como el tamaño, color y posición en la página.

El último botón es diferente y lleva a una URL de FTP en lugar de una página web. Cuando se hace clic en este botón, se conectará a un servidor FTP en la dirección IP 192.168.1.4 y se mostrarán los archivos de registro disponibles. La etiqueta style se utiliza para establecer la posición del botón en la página.

Ahora se explica el archivo CSS que contiene los diferentes estilos:


```
body{
  background: #0F2027; /* fallback for old browsers */
  background: -webkit-linear-gradient(to left, #2C5364, #203A43, #0F2027); /* Chrome 10-25, Safari 5.1-6 */
  background: linear-gradient(to left, #2C5364, #203A43, #0F2027); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+ */
}

h1{
  font-weight: bold;
  text-align: center;
  color: antiquewhite;
}

.barra {
  margin: auto;
  width: 300px;
  height: 20px;
  background-color: #ddd;
  border-radius: 10px;
}

.relleno {
  height: 100%;
  border-radius: 10px;
  background-color: #4CAF50;
}
```

```
.relleno {
  height: 100%;
  border-radius: 10px;
  background-color: #4CAF50;
}

.boton:hover {
  background-color: black;
  color: white;
}

button {
  width: 200px;
  height: 50px;
  font-size: 18px;
  border-radius: 10px;
}

.CP{
  position: absolute;
  bottom: 0;
  right: 0;
  color: rgba(0,0,0,0.7)
}

.Datos{
  color: antiquewhite;
}
```

La primera sección del código establece un degradado de fondo en el cuerpo de la página web, que va desde un color azul oscuro (#2C5364) a un tono más claro (#0F2027). La segunda sección define el estilo para los títulos (h1) en la página web, con un peso de fuente en negrita, un color en tono blanco antiguo y un alineamiento centrado.

Las siguientes secciones establecen el estilo para elementos específicos de la página, como una barra de progreso (clase "barra") que tiene un ancho de 300 píxeles, una altura de 20 píxeles y un borde redondeado de 10 píxeles. La barra tiene un relleno (clase "relleno") que se muestra en un color verde (#4CAF50).

También hay estilos para los botones de la página (clase "boton"), con un fondo blanco y letras en tamaño de fuente de 18 píxeles y un borde redondeado de 10 píxeles. Cuando el cursor pasa sobre un botón, cambia el color de fondo a negro y las letras se vuelven blancas.

Además, hay una clase para elementos de datos (clase "Datos") que establece el color del texto en un tono de blanco antiguo y otra para la posición del pie de página de la página web (clase "CP"), que se establece en la posición absoluta en la esquina inferior derecha de la página y tiene un color de texto semi-transparente en negro.

11.2. Páginas secundarias

Las diferentes páginas que muestran la información de los servicios comparten los mismos estilos que son los siguientes:

```
table {
  border: 3px solid black;
  border-collapse: collapse;
  margin-top: 25px;
  margin-left: auto;
  margin-right: auto;
}

th, td {
  font-size: 20px;
  border: 3px solid black;
  padding: 15px;
}

th {
  color: antiquewhite;
}

td {
  color: white;
}

table td {
  text-align: center;
}

h1 {
  font-size: 25px;
  color: antiquewhite;
  text-align: center;
  margin-top: 50px;
}
```

La primera sección "table" establece que la tabla tendrá un borde sólido de 3 píxeles de ancho en color negro, que el borde se colapsará (no habrá espacios entre las celdas), y que la tabla estará centrada horizontalmente en la página con un margen superior de 25 píxeles.

La segunda sección "th, td" establece que tanto los encabezados de columna como las celdas de la tabla tendrán un tamaño de letra de 20 píxeles, un borde sólido de 3 píxeles de ancho en color negro y un relleno de 150 píxeles (esto hace que las celdas se expandan mucho).

La tercera sección "th" establece que el texto de los encabezados de columna será de color antiquewhite.

La cuarta sección "td" establece que el texto de las celdas de la tabla será de color blanco.

La quinta sección "table td" establece que el texto en las celdas de la tabla estará centrado.

La última sección "h1" establece que los títulos de la página serán de tamaño 25 píxeles, de color antiquewhite, y estarán centrados horizontalmente con un margen superior de 50 píxeles.

Todas las páginas secundarias generan tablas con la información de las diferentes tablas siguiendo el siguiente código PHP:

```
<h1>Usuarios directorio Activo (AD)</h1>
<?php
    // Establecer conexión
    $conn = mysqli_connect("192.168.1.4", "Proyecto", "Admin123.", "INFO_AD");
    $query = "SELECT * FROM Usuarios_AD";
    $result = mysqli_query($conn, $query);

    echo "<table>";
    echo "<tr><th>Nombre Distinguido</th><th>Nombre</th><th>Nombre Principal</th></tr>";

    while ($row = mysqli_fetch_assoc($result)){
        echo "<tr><td>".$row["Nombre_Distinguido"]."</td><td>".$row["Nombre"]."</td><td>".$row["Nombre_Principal"]."</td></tr>";
    }

    echo "</table>";

    mysqli_close($conn);
?>
```

Este código se encarga de recuperar la información de los usuarios del directorio activo y de mostrarla en una tabla HTML. En primer lugar, se establece una conexión con la base de datos de información de Active Directory a través de la función `mysqli_connect()`, utilizando la dirección IP del servidor (192.168.1.4), el nombre de usuario (Proyecto) y la contraseña (Admin123.). Luego, se realiza una consulta SQL mediante la cadena `$query` para recuperar todos los datos de la tabla `Usuarios_AD` y se almacena en la variable `$result` utilizando la función `mysqli_query()`.

Después, se muestra la tabla HTML utilizando la función `echo()`. La tabla tiene un encabezado con tres columnas: "Nombre Distinguido", "Nombre" y "Nombre Principal", que se definen en la primera fila mediante la función `echo()`. Luego, se usa un bucle `while` para iterar sobre los resultados de la consulta `$result`, y se muestra cada fila de la tabla utilizando la función `echo()`. Finalmente, se cierra la conexión con la base de datos usando `mysqli_close()`.

Las únicas cosas que cambian son el encabezado y la tabla que se analiza además de obviamente lo que se muestra. Una vez ejecutado la página se ve así:

Usuarios directorio Activo (AD)

Nombre Distinguido	Nombre	Nombre Principal
CN=Administrador,CN=Users,DC=proyecto,DC=net	Administrador	
CN=Diego,CN=Users,DC=proyecto,DC=net	Diego	Diego@proyecto.net
CN=Invitado,CN=Users,DC=proyecto,DC=net	Invitado	
CN=krbtgt,CN=Users,DC=proyecto,DC=net	krbtgt	
CN=Prueba,CN=Users,DC=proyecto,DC=net	Prueba	Prueba@proyecto.net

Conclusión

Tras llevar a cabo este proyecto, se han logrado satisfactoriamente los objetivos planteados inicialmente. En primer lugar, se ha diseñado una red de servidores dividida en dos sites con diferentes servicios y sistemas operativos. Se ha automatizado la toma de información de estos servicios mediante la creación de scripts de Powershell, y se ha guardado esta información en una base de datos MySQL.

Además, se ha creado una página web que permite el acceso a esta información, proporcionando una forma sencilla y rápida de consultar los datos de los diferentes servicios. Todo ello ha sido posible gracias al uso de diferentes herramientas como XAMPP, FileZilla Server, y el programador de tareas de Windows.

Finalmente, se ha realizado una planificación adecuada del proyecto mediante el uso de un diagrama de Gantt, lo que ha permitido una distribución eficiente del tiempo y recursos.

En conclusión, este proyecto ha permitido poner en práctica los conocimientos adquiridos en el grado superior de sistemas y redes, así como enriquecerlos con nuevas habilidades y herramientas. El resultado final es una red de servidores automatizada y accesible que cumple con los objetivos planteados inicialmente.

Acceso a los recursos del proyecto

Todo el trabajo realizado en este proyecto, incluyendo los scripts, capturas utilizadas y documentación, se encuentra disponible en un repositorio de GitHub en el siguiente enlace: <https://github.com/HypnoticX11/ProyectoGradoSuperior>

Desde allí, cualquier persona interesada en este proyecto puede acceder a todo el contenido y seguir las instrucciones para reproducir los resultados. El acceso al repositorio es público y gratuito, y se puede clonar o descargar para acceder a los diferentes archivos y carpetas que componen este proyecto.

Bibliografía

- Apache Software Foundation (2023) *Documentación Apache*, *Apache docs*. Available at: <https://httpd.apache.org/docs/>.
- *Documentación PHP* (2023) *PHP docs*. Available at: <https://www.php.net/docs.php>.
- *Documentación PowerShell* (2023) *Docu PowerShell*. Microsoft. Available at: <https://learn.microsoft.com/es-es/powershell/>.
- *Documentación web* (2023) *Documentation for Web developers*. Mozilla. Available at: <https://developer.mozilla.org/en-US/docs>.
- Mozilla (2023) *Documentación IPFire*, *IPfire Wiki*. Available at: <https://wiki.ipfire.org/>.
- MySQL (2023) *Documentación MySQL*, *MySQL doc*. Available at: <https://dev.mysql.com/doc/>.