LaVEREDA | ESO BATXILLERAT CICLES

Infraestructura IT para PYMES con Virtualización, Contenedores y Seguridad Integral

Alumno: Juan José Castillo Rodríguez
Centro: IES La Vereda Ciclo: 2º ASIR
Curso académico: 2024–2025
Tutora: Andrea Jordán Jordán

RESUMEN

Resumen:

Este proyecto consiste en el diseño e implementación de una infraestructura IT para una empresa simulada (IPVCSI), dedicada a la creación y gestión de contenido multimedia.

La infraestructura se basa en virtualización, contenedores Docker, alta disponibilidad, seguridad y automatización. Se ha creado una red simulada con dos sedes: Castellón (en la cual estará situada toda la empresa físicamente, con los dos servidores de forma presencial y todos los equipos de la empresa y puestos de trabajo), y Madrid, (que almacenará una réplica exacta de los servidores por seguridad, la cual sirve para que en el caso de que suceda algo en la sede de Castellón la empresa siga teniendo servicio mediante la sede de Madrid). Dichas sedes estarán conectadas mediante VPN, los servidores de esta empresa contarán con un Ubuntu Server (máquina que ofrece todos los servicios a la empresa) y un Windows Server (maquina principal, y controlador de dominio). Entre los servicios destacan aquellos orientados al almacenamiento y distribución de contenido (Nextcloud, Plex), así como la gestión centralizada de usuarios (Active Directory), copias de seguridad y acceso remoto seguro (R.D.P, Duplicati, Wireguard).

Resum:

Aquest projecte consisteix en el disseny i la implementació d'una infraestructura IT per a una empresa simulada (IPVCSI), dedicada a la creació i gestió de contingut multimèdia.

La infraestructura es basa en virtualització, contenidors Docker, alta disponibilitat, seguretat i automatització. S'ha creat una xarxa simulada amb dues seus: Castelló (en la qual estarà situada tota l'empresa físicament, amb els dos servidors de forma presencial i tots els equips de l'empresa i llocs de treball), i Madrid, (que emmagatzemarà una rèplica exacta dels servidors per seguretat, la qual serveix perquè en el cas que succeeixi alguna cosa a la seu de Castelló l'empresa segueixi tenint servei). Aquestes seus estaran connectades mitjançant VPN, els servidors d'aquesta empresa comptaran amb un Ubuntu Server (màquina que ofereix tots els serveis a l'empresa) i un Windows Server (màquina principal i controlador de domini). Entre els serveis destaquen aquells orientats a l'emmagatzematge i la distribució de contingut (Nextcloud, Plex), així com la gestió centralitzada d'usuaris (Active Directory), còpies de seguretat i accés remot segur (R.D.P, Duplicati, Wireguard).

Summary:

This project involves the design and implementation of an IT infrastructure for a simulated company (IPVCSI) dedicated to the creation and management of multimedia content.

The infrastructure is based on virtualization, Docker containers, high availability, security, and automation. A simulated network has been created with two locations: Castellón (where the entire company will be physically located, with both servers and all company equipment and workstations), and Madrid (which will store an exact replica of the servers for security purposes, ensuring that in the event of an incident at the Castellón headquarters, the company will continue to have service through the Madrid headquarters). These locations will be connected via VPN; the company's servers will include an Ubuntu Server (the machine that provides all services to the company) and a Windows Server (the main machine and domain controller). Among the services, those focused on content storage and distribution (Nextcloud, Plex) stand out, as well as centralized user management (Active Directory), backups and secure remote access (R.D.P, Duplicati, Wireguard).

Índice:

1. Parte empresarial

1.1 ESTUDIO PREVIO DE LA EMPRESA

Nombre de la empresa: IPVCSI (Infraestructura Profesional Virtualizada y Cibersegura S.L.)

Forma jurídica: Sociedad Limitada (S.L.)

Ubicación simulada: Castellón de la Plana (sede principal), Madrid (sede secundaria)

Actividad principal: Creación y gestión de contenido multimedia para empresas. Su labor incluye la producción de vídeos, edición de clips, gestión y organización de archivos digitales, y la entrega de materiales finales listos para la publicación y promoción de cada cliente.

Misión: Ofrecer soluciones digitales creativas y seguras, permitiendo a los clientes contar con contenido multimedia de calidad que impulse su imagen corporativa.

Visión: Ser un referente local en la producción de contenido audiovisual personalizado, con procesos internos bien organizados que permitan eficiencia, calidad y confidencialidad en cada entrega.

Valores:

- Compromiso con la calidad y originalidad de cada proyecto.
- Adaptabilidad y mejora constante en la producción y la técnica.
- Seguridad y privacidad en el tratamiento de los datos del cliente.
- Colaboración y comunicación efectiva entre los departamentos internos y con los clientes.

1.1.1 DAFO de la empresa:

Fortalezas	**Debilidades**
– Infraestructura modular, fácil de escalar – Capacidad de trabajo colaborativo entre departamentos – Servicios adaptados a contenido multimedia – Personal técnico cualificado – Oportunidades – Demanda de servicios gestionados – Aumento de la digitalización	– Dependencia inicial de hardware básico / Limitado capital inicial – Necesidad de un mantenimiento continuo – Falta de personal técnico especializado al inicio
– **Oportunidades**	– **Amenazas**
– Mayor demanda de servicios multimedia y audiovisuales – Digitalización de pequeñas empresas – Aumento del trabajo remoto y uso de VPN	– Competencia con grandes proveedores / plataformas SaaS – Ciberataques o vulnerabilidades en la red – Cambios en las regulaciones de protección de datos

TABLA 1. DAFO

1.1.2 Canvas de negocio (resumen):

FIGURA 1. Modelo Canvas

Elemento	Contenido
Segmento de clientes	Empresas o PYMEs que necesitan contenido audiovisual: asesorías, agencias, comercios con necesidades de vídeos o presentaciones
Propuesta de valor	Creación de contenido multimedia profesional, con colaboración fluida entre los equipos de edición y distribución, asegurando calidad y plazos ajustados
Canales	Reuniones directas con clientes, web corporativa, presentación de portfolios digitales y contacto en eventos locales
Relación con clientes	Relación personalizada y seguimiento de cada proyecto; comunicación constante para adaptarse a sus necesidades y cambios durante la producción
Ingresos	Tarifas por proyecto (vídeos, clips, presentaciones) y paquetes de servicio de mantenimiento del contenido digital
Recursos clave	Equipos técnicos especializados (editores, diseñadores, personal de IT), software de edición, servidores y red VPN interna
Actividades clave	Grabación y edición de contenido, almacenamiento y gestión de archivos, revisión y aprobación con el cliente final
Socios clave	Proveedores de material audiovisual, proveedores de energía, asesores legales para derechos de autor y licencias
Costes	Licencias de software de edición, consumo eléctrico de servidores, honorarios del equipo creativo y costes de almacenamiento en red

TABLA 2. Modelo Canvas

1.2 INTRODUCCIÓN

El presente proyecto tiene como objetivo diseñar y desplegar una infraestructura tecnológica completa para una empresa dedicada a la producción y gestión de contenido multimedia, garantizando la colaboración fluida entre los distintos departamentos creativos y la disponibilidad de los recursos digitales.

Para ello se han utilizado distintos servidores, uno de ellos Windows Server para facilitar la gestión de usuarios, objetos de políticas de grupo, y otros recursos relacionados y un servidor Ubuntu Server con el uso de contenedores Docker para servicios colaborativos, medidas de seguridad como VPN (WG-Easy), NextCloud, Duplicati , y mecanismos de alta disponibilidad y recuperación ante desastres.

La empresa simulada se denomina IPVCSI, con sede principal en Castellón y una sede de respaldo simulada en Madrid. Se ha optado por una red enrutada mediante pfSense para garantizar la seguridad y el control del tráfico, y se ha implementado una estructura basada en servicios virtualizados como Nextcloud y Plex, orientados a facilitar la colaboración entre departamentos de edición, distribución y almacenamiento de contenido audiovisual.

1.2.1 Planificación y temporalización

Para organizar el trabajo, se ha estructurado el proyecto en varias fases, cada una con sus tareas y duración estimada:

Fase	Tareas principales	Duración estimada
Análisis inicial	Estudio de necesidades, DAFO, planificación	1 semana
Diseño de red	Topología, direccionamiento, esquema de sedes	1 semana
Preparación de VMs	Instalación de sistemas operativos y pfSense	1 semana
Despliegue de servicios	Docker, Nextcloud, Plex, etc.	2 semanas
Seguridad y HA	VPN, firewall, controladores replicados	1 semana
Evaluación	Pruebas, simulación de fallos, copias	1 semana
Documentación	Redacción del proyecto, esquemas, capturas	2 semanas

TABLA 3. Planificación y temporalización del proyecto

Fase	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8
Análisis inicial								
Diseño de red								
Preparación de VMs								
Despliegue de servicios								
Seguridad y HA								
Evaluación								
Documentación								

TABLA 1. Planificación y temporalización del proyecto

1.2.2 Decisión técnica:

Abandono de Proxmox

Inicialmente se planteó utilizar Proxmox VE como plataforma de virtualización. Sin embargo, debido a problemas de compatibilidad con los adaptadores de red en máquinas virtuales dentro de VirtualBox, se optó por desechar esta opción. Además, los equipos del aula del centro no cuentan con la potencia ni la memoria RAM necesarias para ejecutar múltiples máquinas virtuales bajo Proxmox de manera fluida, lo que comprometía la viabilidad del entorno de prácticas ya que necesitaba crear una maquina virtualizada con proxmox con 8 gb de ram, 4 núcleos, y en su interior dos máquinas virtualizadas más cada una de ellas con 4 gb de ram y 2 núcleos, además de toda la configuración interna que ello conlleva. Por estos motivos, se decidió utilizar directamente máquinas virtuales independientes sobre VirtualBox, una solución más ligera, estable y compatible con el entorno académico.

Descarte de idea de Waulthwarden

Finalmente, se decidió no aplicar el contenedor de Vaultwarden, ya que se utilizará la gestión de contraseñas a través de Active Directory. La implementación de Vaultwarden se considera una posible mejora futura, orientada a centralizar las contraseñas comunes de los usuarios.

Descarte de OpenMediaVault

Se analizó la posibilidad de utilizar OpenMediaVault como sistema de almacenamiento en red. No obstante, fue descartado al comprobarse que no es compatible con Ubuntu Server 24.04, sistema base utilizado en este proyecto. OMV solo es oficialmente compatible con Debian 11 (Bullseye) y Debian 12 (Bookworm), y no funciona correctamente en distribuciones derivadas como Ubuntu. En su lugar, se eligió Nextcloud como alternativa, ya que permite la gestión y compartición de archivos, cubriendo así las necesidades previstas.

1.3 INFRAESTRUCTURA DEL PROYECTO

En este apartado se explica cómo se ha diseñado la infraestructura de red y qué tecnologías se han utilizado. La idea principal ha sido crear un entorno funcional y seguro para una empresa que se dedica a la creación y gestión de contenido multimedia, con especial atención a la virtualización, los servicios en contenedores, la seguridad y la alta disponibilidad.

1.3.1 Topología de red

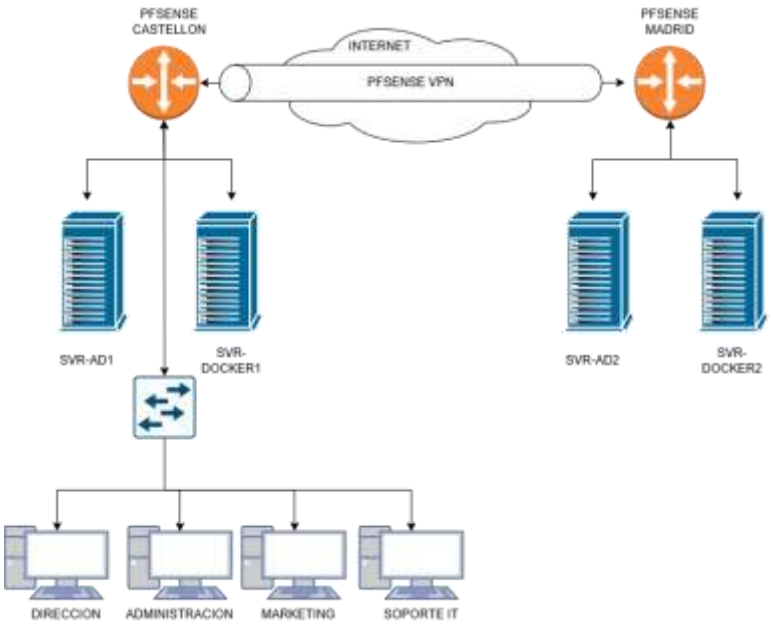


FIGURA 2. Topología de red entre sedes (Castellón y Madrid)

Se ha simulado una red empresarial con dos sedes:

- **Sede principal:** Castellón – Red 10.20.10.0/24
- **Sede secundaria:** Madrid – Red 10.30.10.0/24

Ambas sedes están conectadas mediante una VPN site-to-site usando WireGuard, configurada en pfSense. La sede de Castellón es la principal, y desde allí se gestiona el DHCP, los servidores principales y toda la operativa de la empresa. En Madrid simplemente se encuentran los servidores de respaldo, que estarán siempre sincronizados y preparados por si ocurre un fallo en Castellón.

1.3.2 Virtualización

Se han utilizado máquinas virtuales en VirtualBox. La elección de VirtualBox se debe a que es una solución ligera, gratuita y totalmente funcional para el entorno de prácticas. Las máquinas virtuales se han dividido de la siguiente manera:

Nombre	Función	IP
srv-ad	Controlador de dominio principal	10.20.10.11
srv-ad2	Controlador de dominio secundario	10.20.10.21
srv-docker	Servidor principal con servicios Docker	10.20.10.12
srv-docker2	Réplica de respaldo (Madrid)	10.20.10.22

TABLA 2. Detalle de VMs utilizadas en el proyecto

- srv-ad y srv-ad2: Windows Server 2019 o 2022
- srv-docker y srv-docker2: Ubuntu Server 22.04 LTS

Cada servidor tiene asignada una IP fija y se encuentran correctamente integrados dentro del dominio ipvcsi.local.

1.3.3 Servidores y servicios desplegados

Los servidores se han dividido en dos roles principales. Por un lado, los controladores de dominio con Windows Server, y por otro, los servidores Ubuntu con servicios en contenedores Docker.

En el servidor Windows Server (srv-ad), se ha instalado el rol de Active Directory y DNS para poder gestionar los usuarios de la empresa, aplicar políticas y controlar todo lo relacionado con el dominio. Este servidor es el centro de la gestión interna. En la sede secundaria (Madrid) se ha montado un segundo servidor (srv-ad2) como copia de seguridad, que replica todos los datos del principal por si ocurre algún fallo.

En los servidores Ubuntu (srv-docker y srv-docker2), se han desplegado distintos servicios usando contenedores Docker. Estos servicios permiten a la empresa almacenar archivos, gestionar contenido, realizar copias de seguridad, acceder

desde fuera por VPN, y tener un entorno multimedia para el departamento de marketing.

Además, se ha instalado Portainer para poder controlar todos los contenedores desde una interfaz gráfica sin necesidad de usar comandos.

1.3.4 Configuración de red y direccionamiento

Cada sede está en una red interna aislada. pfSense actúa como firewall y gateway. Se ha configurado una red NAT personalizada (10.0.0.0/16) para las interfaces WAN de ambas sedes para que puedan tener comunicación entre ellas, ya que, si elegíamos configuraciones NAT por defecto de cada máquina, esta no permitiría la comunicación entre máquinas.
También se configuro una vpn por defecto en pfsense para crear el túnel que permita que se vean las maquinas entre ellas, la dirección vpn por defecto elegida fue la siguiente:

	UBICACIÓN	IP
RED	VPN	10.99.99.0/24
SEDE	CASTELLON	10.99.99.1/24
SEDE	MADRID	10.99.99.2/24

TABLA 3. Asignación de direcciones IP y funciones por servidor

1.3.5 Sistema de backups y recuperación

Se ha preparado un sistema de copias de seguridad automático para asegurar que, si falla un servidor, los datos estén protegidos.

- En srv-docker, se usa Duplicati, que realiza copias programadas de los contenedores y sus volúmenes.
- Estas copias se guardan en una carpeta compartida de red alojada en srv-ad.
- En srv-ad, también se ha instalado Veeam Agent para hacer copias completas del sistema y del estado de Active Directory.
- Además, en Madrid está desplegado srv-docker2, una réplica que puede asumir el rol de principal si srv-docker falla.

Gracias a este sistema, la empresa puede recuperar su operativa en poco tiempo ante un problema grave.

1.4 SEGURIDAD Y GESTIÓN DE LA INFRAESTRUCTURA

La seguridad de la red se ha gestionado desde varios niveles. Todos los servicios internos solo son accesibles desde la red de la empresa (10.20.10.0/24). Además, la VPN permite conectarse desde fuera de forma segura.

El firewall de pfSense controla el tráfico entre sedes y bloquea accesos no autorizados desde fuera. También se han configurado reglas específicas para los puertos necesarios.

Duplicati realiza las copias de forma automática y cifrada, y se almacenan en una carpeta compartida de red.

2. Fundamentos y conceptos

En este apartado se explican algunas de las herramientas clave utilizadas en el proyecto y por qué se han elegido.

2.1 PfSense

Se ha elegido pfSense como router y firewall de la red porque es una solución libre, potente y muy visual. Permite configurar fácilmente VPNs, reglas de firewall, servidores DHCP y más.

FIGURA 3. Página principal de pfsense

La configuración básica ha sido:

- **pfSense-Castellón:**
  - WAN: 10.0.2.15
  - LAN: 10.20.10.1
- **pfSense-Madrid:**
  - WAN: 10.0.20.15
  - LAN: 10.30.10.1

Ambos se conectan por VPN usando WireGuard, y la red de la VPN es la 10.99.99.0/24. Se ha hecho ping entre los routers y entre las máquinas de ambas sedes para comprobar la conexión.

Objetivo

El objetivo de usar pfSense en este proyecto es tener un cortafuegos virtual que funcione como router, firewall, servidor DHCP y reenviador DNS para cada red. Gracias a esto, se consigue una red interna organizada, con control de acceso a los servicios y con una puerta de enlace bien definida.

pfSense también facilita la creación del túnel VPN entre las dos sedes, algo fundamental para que los servidores puedan comunicarse como si estuvieran en la misma oficina.

2.1.1 DHCP:

En la sede de Castellón se ha activado un servidor DHCP en pfSense, que asigna direcciones IP a los equipos de forma automática dentro del rango 10.20.10.100 a 10.20.10.200.

En la sede de Madrid se ha desactivado el DHCP, ya que todos los dispositivos allí utilizan IP fija o reciben IP a través de la VPN desde Castellón. Esta decisión se ha tomado para tener una gestión centralizada de las direcciones IP, reducir posibles conflictos y facilitar el mantenimiento de la red.

2.1.2 VPN con WireGuard

Para conectar las dos sedes de forma segura, se ha creado una VPN tipo site-to-site usando WireGuard. En este caso:

- pfSense-Castellón actúa como servidor de la VPN.
- pfSense-Madrid actúa como cliente y se conecta automáticamente.

Ambas redes (10.20.10.0/24 y 10.30.10.0/24) quedan conectadas a través de la red de la VPN 10.99.99.0/24. Con esto, los servidores de Madrid (que son copias de los principales) pueden estar siempre sincronizados y listos para entrar en acción si algo falla en la sede principal.

FIGURA 1. Configuración VPN de pfsense

2.1.3 Firewall con pfSense

El firewall de pfSense se ha configurado con reglas específicas para permitir solo el tráfico necesario. Por ejemplo:

- Se permite el tráfico VPN por el puerto 51820 (UDP).
- Se permite el acceso a servicios internos como Nextcloud, Plex, Pi-hole, etc., por los puertos personalizados que se han asignado.
- Se bloquean las conexiones no autorizadas desde el exterior.

También se han creado reglas en la interfaz de la VPN para permitir el tráfico entre ambas sedes, asegurando que todos los servicios puedan comunicarse correctamente.

2.1.4 Routing

Para que las redes de las dos sedes puedan comunicarse sin problemas, se han creado rutas estáticas en ambos pfSense. Estas rutas permiten que el tráfico entre las redes 10.20.10.0/24 y 10.30.10.0/24 se dirija correctamente a través de la VPN.

Además, se ha creado una gateway virtual asociada a la interfaz WireGuard (wg0), ya que pfSense no la genera automáticamente. Esta gateway se utiliza para enrutar el tráfico entre sedes.

2.1.5 Configuración DNS

Durante las pruebas iniciales, se detectó que, aunque había conexión a internet, no se resolvían correctamente los nombres de dominio. Esto se debía a un fallo del servicio DNS Resolver de pfSense.

Para solucionarlo, se desactivó el DNS Resolver y se activó el DNS Forwarder. Después se añadieron servidores DNS públicos (8.8.8.8 y 1.1.1.1) como reenviadores. Con esto, ya se resolvían los nombres sin problema y la navegación funcionaba correctamente.

Esta misma configuración se aplicó también en pfSense-Madrid, para garantizar que todo funcione igual en ambas sedes.

2.1.6 Punto crítico de pfSense-Castellón

Un punto importante del sistema es que pfSense-Castellón es el centro de todo. Por él pasa la VPN, el DHCP y todo el tráfico. Si este equipo falla:

- No habría salida a internet.
- No se asignarían IPs.
- Se perdería la conexión entre sedes.

Para evitar esto, se ha creado una copia exacta de pfSense-Castellón con toda su configuración, llamada pfSense-Castellón Backup, preparada por si el principal dejara de funcionar. Además, se podría plantear contratar una segunda línea de internet con otro proveedor para reducir riesgos aún más.

2.1.7 Alta disponibilidad

Para asegurar el funcionamiento del sistema ante fallos, se han tomado varias medidas de alta disponibilidad:

- Se ha creado un segundo controlador de dominio en Madrid (srv-ad2), que replica todos los datos de Active Directory. Si el servidor principal falla, este puede seguir autenticando usuarios y resolviendo DNS sin problema.
- También se ha creado una réplica del servidor Docker (srv-docker2), que recibe los backups de srv-docker. En caso de fallo, srv-docker2 puede activarse y seguir prestando servicio.

Con estas medidas, se garantiza que la empresa pueda seguir funcionando aunque falle un servidor importante.

2.1.8 Buenas prácticas de red

Para mejorar la seguridad y el control del entorno, se han aplicado varias buenas prácticas:

- Segmentación de redes por sede.
- Documentación de todas las IPs y servicios.
- Uso de DNS interno con Pi-hole para filtrar publicidad y dominios sospechosos.
- Todo el tráfico entre sedes está cifrado por la VPN.
- Copias de seguridad automáticas y monitorización con Zabbix.

Además, se han cambiado los puertos por defecto de los servicios más comunes para evitar ataques automáticos que escanean puertos típicos. Esto añade una capa extra de seguridad.

2.2 Servidor Windows server

En este proyecto se ha utilizado Windows Server para centralizar la gestión de usuarios, grupos y políticas dentro del dominio ipvcsi.local. Gracias a Active Directory, se puede controlar quién accede a qué recursos, aplicar configuraciones de grupo y tener una organización clara por departamentos.

El servidor principal (srv-ad) está en la sede de Castellón y actúa como controlador de dominio. En la sede de Madrid se ha desplegado otro servidor (srv-ad2), que es una réplica del primero para asegurar la continuidad del servicio en caso de fallo.

2.2.1 GESTIÓN DE USUARIOS

Active Directory

Se ha configurado srv-ad como controlador de dominio y se ha creado la estructura de unidades organizativas (OUs) para representar los distintos departamentos de la empresa: marketing, administración, soporte IT y dirección.

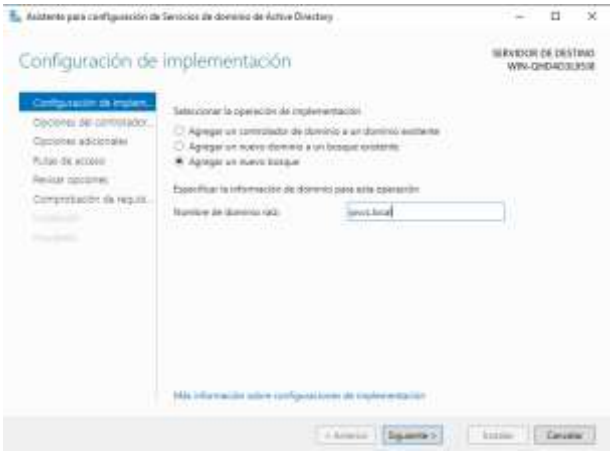


FIGURA 2. Introducción del controlador de dominio al dominio

Dentro de cada OU se han creado usuarios de prueba con nombres reales simulados, como por ejemplo:

- jlopez – Juan López – Marketing
- acastro – Ana Castro – Administración
- rruiz – Raúl Ruiz – Soporte IT
- cgarcia – Carmen García – Dirección

También se han creado grupos de seguridad por departamento:

- G_Marketing
- G_Administracion
- G_Soporte IT
- G_Direccion

Esto facilita aplicar políticas de grupo específicas según el área.

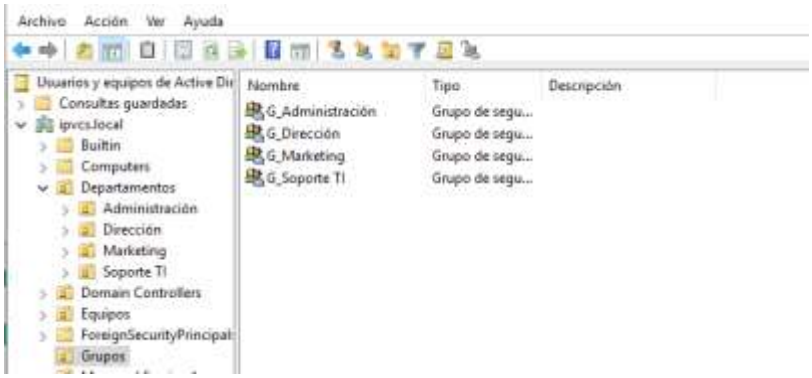


FIGURA 3. Creación del grupos, usuarios y unidades organizativas

Además, se han asignado funciones específicas a cada usuario, en función del departamento y los servicios a los que deben acceder. Esto permite simular un entorno realista donde cada trabajador tiene acceso solo a las herramientas necesarias para su trabajo diario. La siguiente tabla resume las funciones:

Usuario	Nombre completo	OU / Departamento	Función asignada principal
jlopez	Juan López	Marketing	Acceso completo a Plex para subir, organizar y reproducir contenido multimedia.
acastro	Ana Castro	Administración	Acceso a Nextcloud para subir facturas, nóminas y documentación contable.
rruiz	Raúl Ruiz	Soporte IT	Acceso total a todos los servicios, incluido Portainer y Zabbix. Administra copias, VPN y Pi-hole.
cgarcia	Carmen García	Dirección	Acceso a Nextcloud y Plex en modo lectura. Puede consultar informes y vídeos corporativos, pero no modificarlos.

TABLA 4. Usuarios, grupos y funciones que puede realizar cada uno de ellos

Instalar y configurar el servidor DNS en srv-ad

El mismo servidor srv-ad también actúa como servidor DNS interno. Esto es necesario para que los equipos del dominio puedan localizar servicios como el controlador de dominio, carpetas compartidas, etc.

Se ha configurado una zona directa ipvcsi.local y se han creado los registros A necesarios para los servidores. También se ha añadido reenviadores DNS hacia Pi-hole y servidores públicos como 8.8.8.8 para resolver direcciones externas.

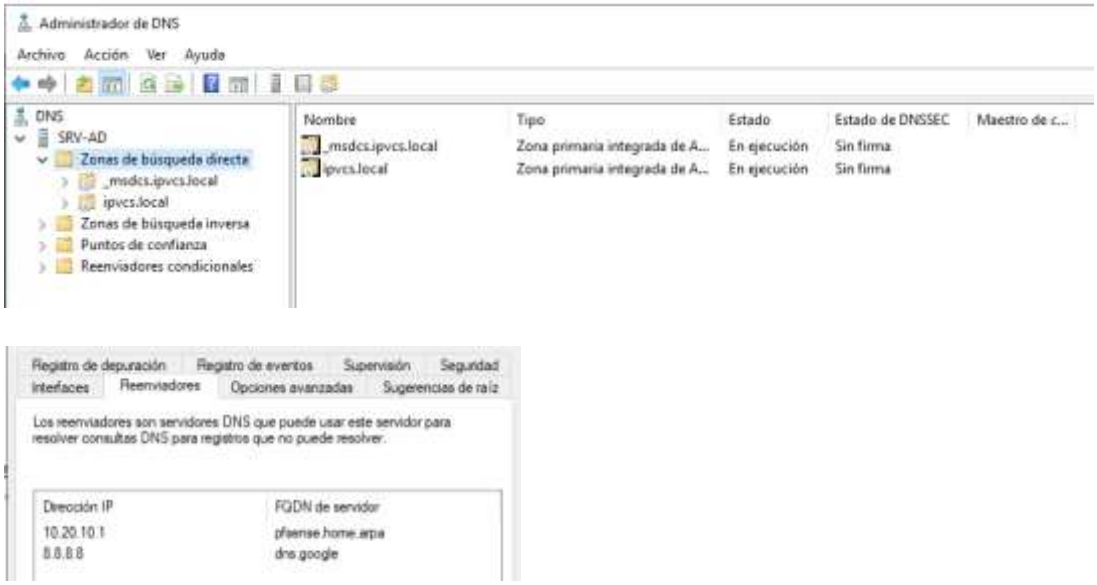


FIGURA 4. Configuración de dns del controlador de dominio

Crear usuarios, grupos y aplicar GPOs en srv-ad

Se han aplicado políticas de grupo básicas para simular un entorno real de empresa. Algunas de las configuraciones que se han implementado son:

- Cambio del fondo de pantalla corporativo para todos los usuarios del dominio, a través de una GPO que apunta a una imagen en una carpeta compartida.
- Mapeo de una unidad de red para que los usuarios tengan acceso a recursos compartidos automáticamente al iniciar sesión.

- Organización por OUs para aplicar políticas diferentes según el departamento.

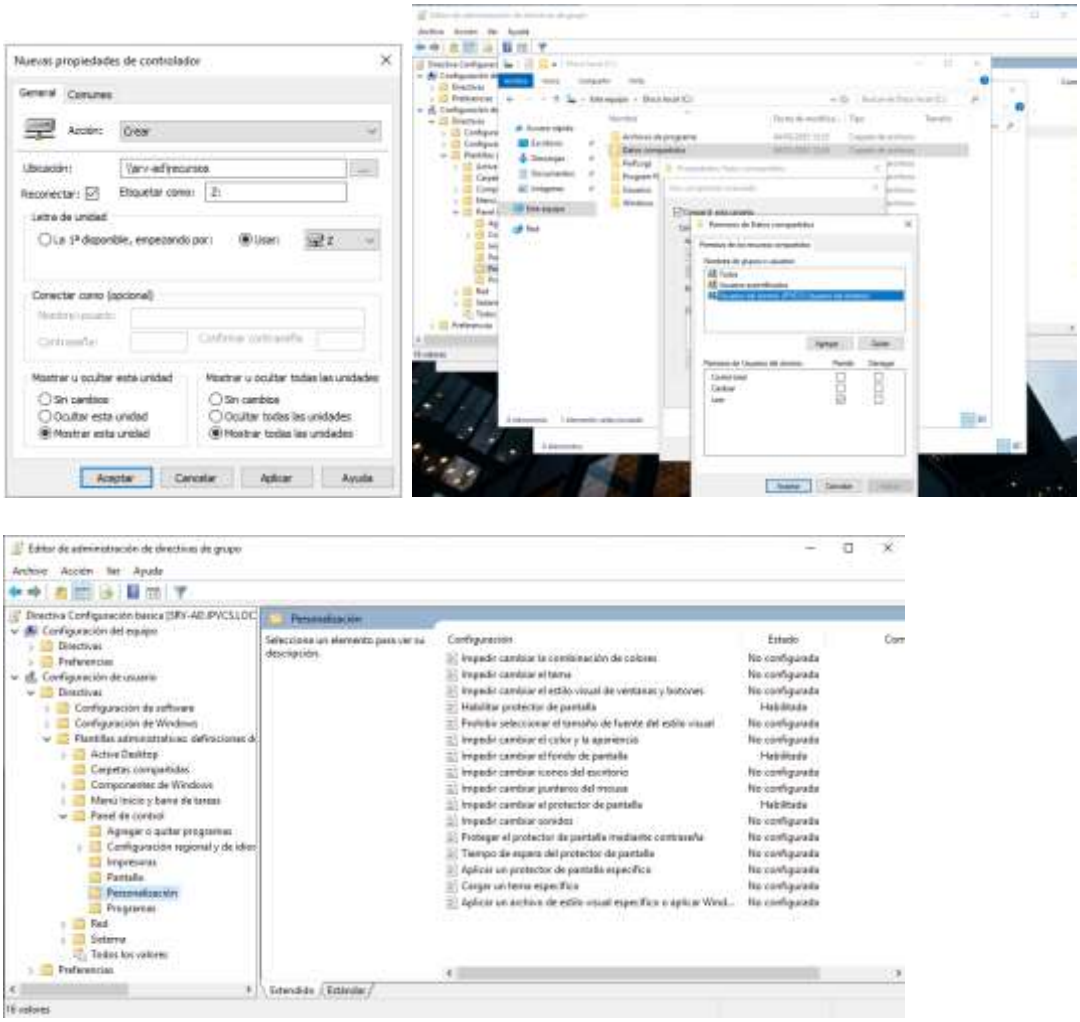


FIGURA 5. Diversas configuraciones de gpo

Configurar NTP en srv-ad1 (controlador principal)

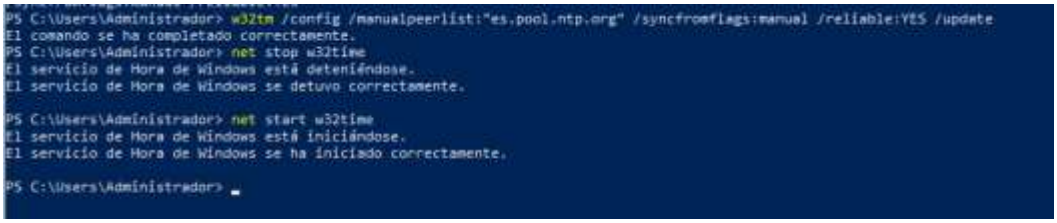
Se ha configurado srv-ad para que sincronice la hora con un servidor NTP externo (por ejemplo pool.ntp.org). Esto es importante para que todos los equipos del dominio tengan la misma hora, ya que es un requisito básico para que funcionen los servicios como la autenticación.

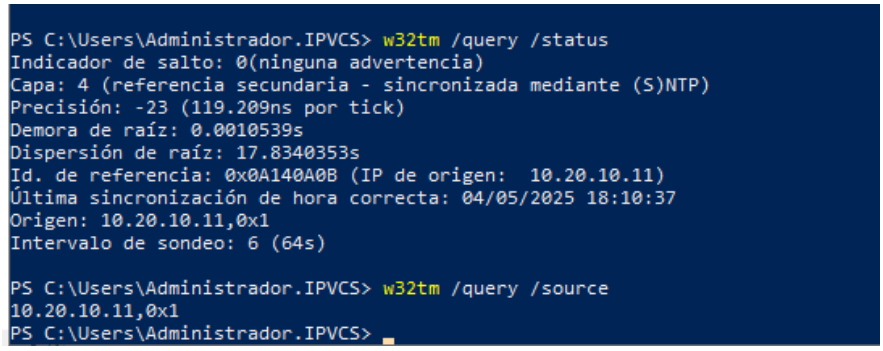
```
Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

PS C:\Users\Administrador> w32tm /config /manualpeerlist:"es.pool.ntp.org" /syncfromflags:manual /reliable:YES /update
El comando se ha completado correctamente.
PS C:\Users\Administrador> net stop w32time
El servicio de Hora de Windows está deteniéndose.
El servicio de Hora de Windows se detuvo correctamente.

PS C:\Users\Administrador> net start w32time
El servicio de Hora de Windows está iniciándose.
El servicio de Hora de Windows se ha iniciado correctamente.

PS C:\Users\Administrador>
```

FIGURA 6. Configuración NTP

Configurar NTP en srv-ad2 (controlador secundario)

El servidor secundario srv-ad2 se ha configurado para que sincronice la hora con srv-ad, y si este falla recurra al mismo servidor NTP externo, pool.ntp.org. De esta forma, si algún día el servidor principal falla, srv-ad2 seguirá funcionando con la misma configuración de hora.





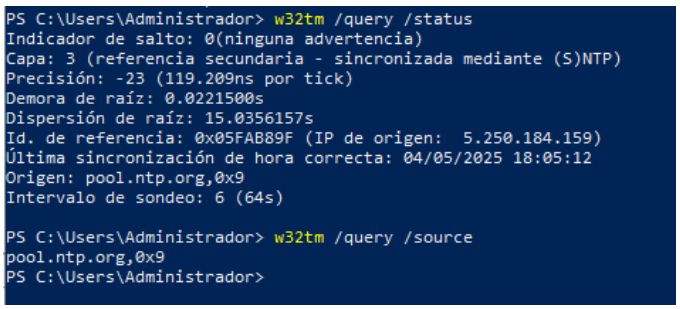


FIGURA 7. Configuración NTP segundo controlador d e dominio

Crear una GPO específica para clientes del dominio

Se ha creado una GPO adicional para aplicar configuraciones solo a los equipos cliente del dominio, sin afectar a los servidores. Para ello, se creó una unidad organizativa (OU) específica donde se movieron las máquinas cliente, y sobre esa OU se aplicó la política.

Dentro de esta GPO se configuró la sincronización de hora (NTP) para que los clientes tomen como referencia el controlador de dominio (srv-ad) y, si este no está

disponible, usen srv-ad2. Así se garantiza que todos los equipos tengan siempre la hora correcta, algo fundamental para que el dominio funcione sin errores.

FIGURA 8. Configuración GPO para clientes del dominio

2.2.2 Controlador secundario

El servidor srv-ad2, ubicado en la sede de Madrid, se ha configurado como controlador de dominio adicional. Esto permite que se replique toda la información de Active Directory desde el servidor principal.

Gracias a esto, si el servidor srv-ad dejase de funcionar, srv-ad2 podría seguir autenticando a los usuarios y resolviendo DNS sin problema. Esto mejora la disponibilidad del sistema y evita que un fallo en el servidor principal afecte al funcionamiento general de la empresa.

También se han hecho pruebas para comprobar que la replicación entre ambos servidores funciona correctamente, y que srv-ad2 puede tomar el control temporal si es necesario.

Como medida adicional, se ha creado un documento de emergencia paso a paso para restaurar el funcionamiento del dominio en caso de caída prolongada de srv-ad1. Este procedimiento permite transferir los roles FSMO de forma forzada a srv-ad2 y volver a transferirlos cuando el servidor principal esté operativo.

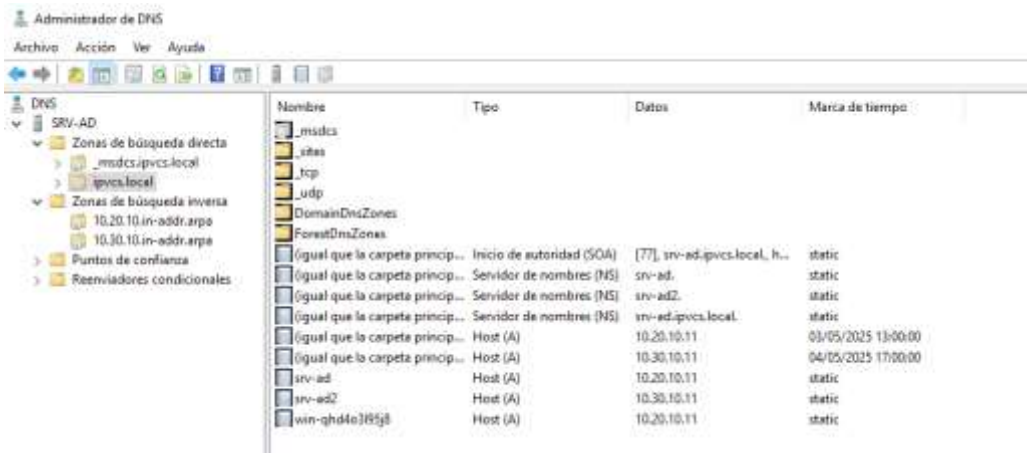
El documento completo puede consultarse en el Anexo 8.

2.2.3 Promocionar srv-ad2 como Controlador de Dominio Adicional de ipvcs.local

Para que srv-ad2 pueda actuar como respaldo completo del controlador de dominio, se ha unido al dominio ipvcsi.local y se ha promovido como controlador de dominio adicional.

Durante este proceso se comprobó que la conexión con srv-ad era correcta, que el DNS funcionaba, y que los tiempos estaban sincronizados. Una vez añadido el rol de Active Directory, se siguió el asistente para unirlo al dominio como DC adicional.

Después del reinicio, se confirmó que los usuarios, OUs y grupos creados en srv-ad también aparecían en srv-ad2, lo que demuestra que la replicación estaba funcionando correctamente. También se probó que se podían autenticar usuarios desde este servidor si el principal no estaba disponible.



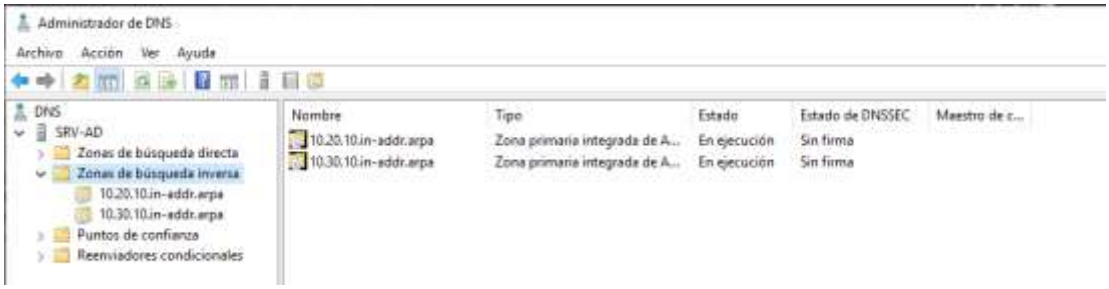


FIGURA 9. Configuración DNS del segundo controlador de dominio

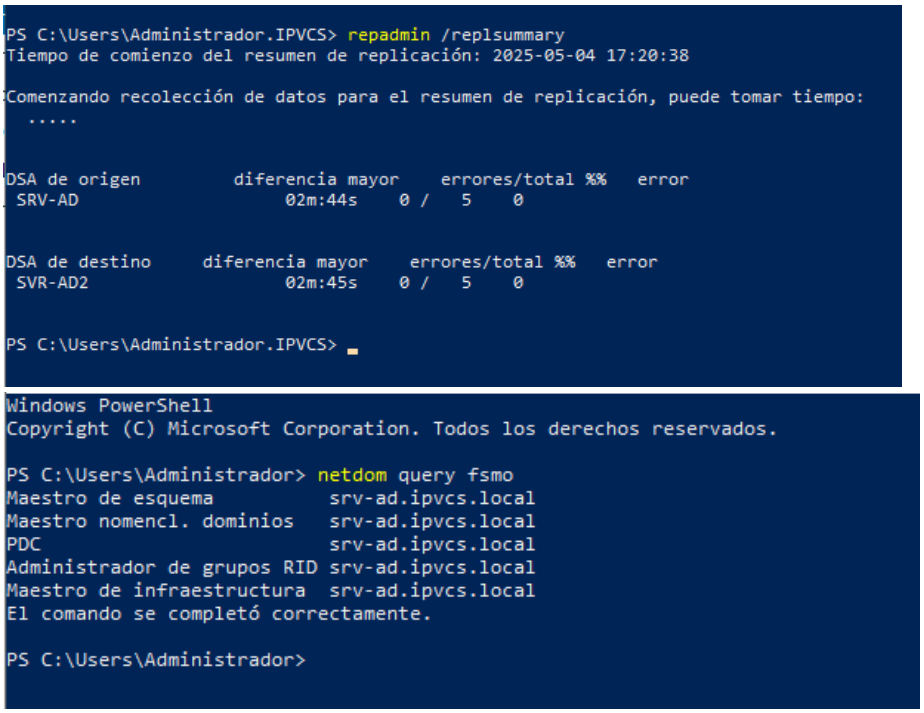


FIGURA 10. Comprobación de roles FSMO del controlador de dominio

2.2.4 Integración con servicios

En este proyecto no se ha aplicado la integración de Active Directory con otros servicios (como Nextcloud o Vaultwarden) a través de LDAP, pero es algo que podríamos aplicar.

Esta integración permitiría que los usuarios usen sus mismas credenciales del dominio para iniciar sesión en servicios como Nextcloud, evitando tener que crear cuentas duplicadas. Se considera una mejora interesante a tener en cuenta si se quiere ampliar el proyecto más adelante.

2.2.5 Activación de backups automáticos con Veeam Agent

Se ha instalado Veeam Agent for Windows en srv-ad y srv-ad2 para realizar copias completas de los servidores. Esto incluye el sistema operativo, las configuraciones y el estado de Active Directory.

La copia se realiza de forma automática todos los días y se guarda en una carpeta compartida en red. También se ha generado una imagen ISO de recuperación, que permite restaurar rápidamente el sistema en caso de desastre. Esta imagen se guarda en una ruta compartida accesible desde otros equipos.

Además, los backups están cifrados y tienen una retención de 7 días, lo que permite recuperar el sistema a distintos puntos si fuera necesario.

FIGURA 11. Información básica sobre Veeam Agent

2.2.6 Alta disponibilidad mediante réplica y backup

Con esta estructura, el sistema es capaz de seguir funcionando aunque uno de los controladores de dominio falle. Los dos servidores (srv-ad y srv-ad2) están siempre sincronizados y preparados para actuar en cualquier momento.

- Si srv-ad se cae, srv-ad2 puede asumir el control.
- Si luego se recupera srv-ad, se puede volver a dejar como servidor principal sin perder datos.

Esto también se aplica al servidor Docker, que tiene su réplica (srv-docker2) y sus datos sincronizados con Duplicati. El sistema está preparado para ser tolerante a fallos de forma básica, sin necesidad de herramientas complejas de clúster.

2.2.7 Simulación de fallo y recuperación

Para comprobar que todo funciona correctamente, se ha simulado una caída del servidor srv-ad. Durante la prueba:

- srv-ad fue apagado por completo.
- Se comprobó que los usuarios podían seguir iniciando sesión gracias a srv-ad2.

- También se verificó que el DNS interno y la red seguían funcionando.

Una vez encendido de nuevo srv-ad, el sistema volvió a su estado normal. Esto demuestra que la alta disponibilidad está bien implementada y preparada para una situación real.

2.3 Servidor docker

El servidor srv-docker es el que se encarga de alojar todos los servicios que la empresa va a utilizar, y lo hace mediante contenedores Docker. Esta forma de trabajar permite tener todos los servicios separados entre sí, pero en una sola máquina, lo que ahorra recursos y facilita mucho la gestión y el mantenimiento.

Se ha instalado Docker y Docker Compose sobre Ubuntu Server 22.04 LTS. También se ha añadido Portainer, que ofrece una interfaz gráfica muy útil para gestionar todos los contenedores desde el navegador, sin necesidad de usar comandos.

El uso de contenedores también permite que si en algún momento hay que mover los servicios a otro servidor (por ejemplo, por fallo de hardware), se pueda hacer de forma rápida. De hecho, todo el contenido de los servicios se guarda en volúmenes, que se pueden copiar y restaurar fácilmente en otra máquina como srv-docker2.

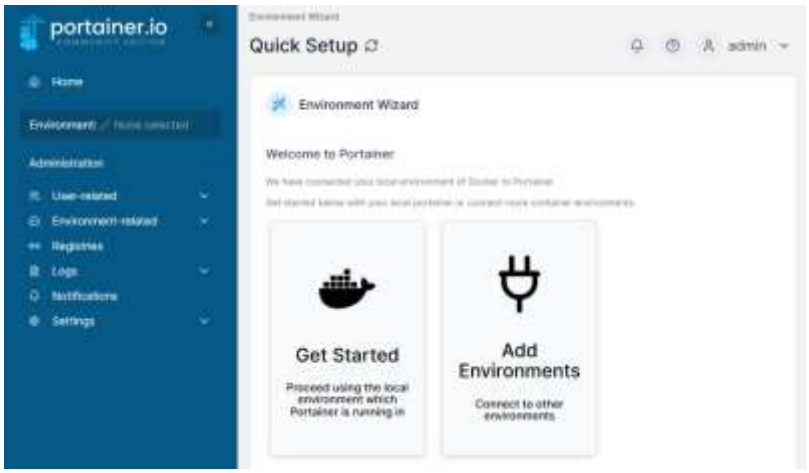


FIGURA 12. Página principal de portainer

Servicios en contenedores (Docker)

Servicios desplegados

Dentro del servidor Docker se han desplegado varios servicios esenciales para la empresa. Cada uno de ellos se ejecuta en un contenedor independiente. Esto permite aislar su funcionamiento y facilita la gestión.

También se ha desplegado un contenedor llamado duckdns, encargado de actualizar automáticamente la IP pública del servidor en el dominio configurado en Duck DNS. Esto permite acceder a los servicios desde fuera sin importar si la IP cambia, ya que el contenedor mantiene actualizada la dirección cada pocos minutos.

Servicio	Función principal
Nextcloud	Plataforma de almacenamiento y colaboración en la nube privada.
Pi-hole	Bloqueador de publicidad y filtrado DNS para toda la red.
Plex	Servidor multimedia para contenido del departamento de marketing.
Duplicati	Sistema de copias de seguridad automáticas cifradas.
Speedtest	Medición automática del rendimiento de la conexión a internet.
Zabbix	Monitorización de servidores y servicios críticos.
MySQL	Base de datos interna para servicios que lo requieran.
WG-Easy	VPN con interfaz gráfica para conectar usuarios de forma segura.
DuckDNS	Actualiza automáticamente la IP pública en el dominio dinámico duckdns.org
Watchtower	Actualización automática de contenedores.
Ansible	Automatización de configuraciones y despliegues.

TABLA 5. Servicios que contiene el servidor Ubuntu y sus funciones

Todos estos servicios están funcionando correctamente en el servidor principal srv-docker, y una copia de seguridad se realiza de forma automática hacia srv-docker2, por si en algún momento hay un fallo en el servidor principal.

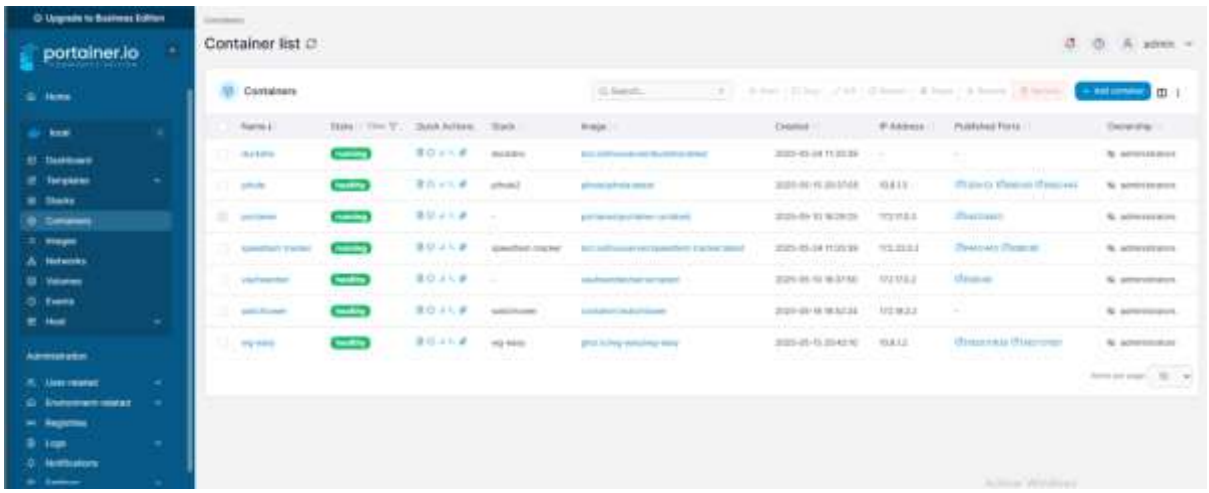


FIGURA 13. Principales contenedores de docker

Puertos personalizados y accesos

Para mejorar la seguridad, se ha decidido no utilizar los puertos por defecto de cada servicio. Esto hace que sea más difícil que alguien desde fuera pueda encontrar qué servicios están activos en la red con un simple escaneo.

A continuación se muestra una tabla que recoge los puertos personalizados asignados a cada servicio, junto con su dirección de acceso desde la red local:

Servicio	Puerto por defecto	Puerto personalizado	Dirección de acceso local
Nextcloud	80/443	8088	http://10.20.10.12
Pi-hole	80/443	8181	http://10.20.10.12:9095/admin
Duplicati	8200	8320	http://10.20.10.12:8200
Speedtest	80/443	8888	http://10.20.10.12:9090
Plex	32400	32400 (sin cambio)	http://10.20.10.12:32400/web
Zabbix	80/443 / 10051	8080 (web) / 10515	http://10.20.10.12:8080
MySQL	3306	3366	(Uso interno, sin acceso web)
WG-Easy VPN	51820/UDP	52828/UDP	Conexión vía cliente WireGuard
DuckDNS	—	—	Sin acceso web – actualiza IP pública
Portainer*	9000 / 9443	9443 (opcional)	http://10.20.10.12:9443

TABLA 6. Contenedores docker, puertos seleccionados y direcciones de acceso

Estos puertos se han configurado tanto en Docker como en pfSense para que el tráfico pase correctamente, y se han añadido las reglas necesarias en el firewall.

El contenedor DuckDNS no tiene una interfaz web, pero es fundamental para mantener actualizada la IP pública del servidor en el dominio dinámico configurado. Gracias a él, se puede acceder remotamente a la VPN y otros servicios aunque la IP cambie, sin necesidad de intervención manual.

Persistencia de datos

Uno de los puntos más importantes al trabajar con contenedores es asegurar que los datos no se pierdan si el contenedor se detiene, se borra o se actualiza. Por eso, se ha configurado volúmenes persistentes en Docker para todos los servicios que manejan datos importantes.

Estos volúmenes están guardados en una carpeta específica del sistema (/srv/volumenes) y se incluyen en las copias de seguridad con Duplicati. Así, si ocurre un problema, se pueden restaurar fácilmente todos los datos de cada contenedor sin necesidad de reconfigurar todo desde cero.

Actualización automática (Watchtower)

Para mantener los contenedores siempre actualizados y corregir posibles fallos de seguridad, se ha instalado el contenedor Watchtower. Este servicio revisa cada cierto tiempo si hay versiones nuevas de los contenedores instalados y, si las encuentra, actualiza automáticamente el contenedor correspondiente sin afectar los datos ni la configuración.

Con esto, se evita tener que estar pendiente de actualizar manualmente cada contenedor y se mejora la seguridad del sistema sin esfuerzo.

Automatización futura con Ansible

Aunque en este proyecto no se ha trabajado directamente con Ansible, se ha dejado preparado el entorno para poder usarlo en el futuro. Ansible permite automatizar tareas repetitivas como instalaciones, actualizaciones o configuraciones dentro de los servidores.

En este caso, se ha instalado Ansible en el servidor Docker para futuras pruebas. Esta herramienta puede ser muy útil si el proyecto se amplía y hay que gestionar más máquinas o servicios al mismo tiempo.

2.4 Copias de seguridad

Duplicati para contenedores

FIGURA 14. Página principal de duplicati

El servicio de copias de seguridad principal en el servidor Docker es Duplicati. Esta herramienta se ejecuta en un contenedor y se encarga de realizar copias cifradas de los volúmenes donde se guardan los datos de cada servicio (como Nextcloud, MySQL o Pi-hole).

Las copias se programan para hacerse todos los días a una carpeta de red compartida, accesible también desde el servidor de respaldo srv-docker2. Con esto se asegura que, en caso de fallo, se pueda restaurar toda la información desde la copia más reciente.

Duplicati también permite elegir el nivel de compresión y el tiempo de retención de las copias, lo que facilita gestionar el espacio en disco.

FIGURA 15. Menú de duplicati

Veeam Agent en controladores de dominio

Para los servidores Windows (srv-ad y srv-ad2) se ha utilizado Veeam Agent, una herramienta gratuita para hacer copias completas del sistema. Esta solución permite hacer backup del estado del sistema operativo, la configuración de Active Directory y todo el disco, creando una imagen completa que se puede restaurar fácilmente.

Las copias se guardan en una carpeta compartida de red, y se ha creado una imagen de recuperación que permite restaurar el sistema desde cero en caso de fallo grave.

También se ha configurado Veeam para enviar un aviso por correo si falla alguna copia, lo que permite detectar problemas a tiempo.

Restauración de copias

Se han hecho pruebas de restauración para comprobar que las copias de seguridad funcionan correctamente. En el caso de los contenedores Docker, se simuló un fallo eliminando un volumen de datos de prueba y restaurándolo desde Duplicati. El resultado fue satisfactorio: los datos se recuperaron sin errores y el servicio volvió a estar operativo.

Para los servidores Windows, se utilizó la imagen de recuperación de Veeam y se arrancó desde ella en VirtualBox. El proceso permitió restaurar todo el sistema, incluyendo Active Directory, tal y como estaba en el momento del backup.

Testeo de restauración

Los test de recuperación se hicieron de forma controlada y programada. Se anotaron los tiempos de restauración y se comprobó que los servicios volvían a funcionar sin problemas. Gracias a estas pruebas se pudo asegurar que, ante un fallo real, la empresa podría recuperar sus sistemas en poco tiempo.

También se verificó que la sincronización entre srv-docker y srv-docker2 era correcta, y que la copia de Duplicati podía restaurarse directamente en el servidor de respaldo si el principal fallaba.

Conclusiones de seguridad y respaldo

El sistema de copias de seguridad implementado es sencillo pero muy efectivo. Se hacen copias automáticas y cifradas, se almacenan en red, y se ha comprobado que se pueden restaurar sin complicaciones.

Esto permite tener tranquilidad ante cualquier fallo, ya sea por error humano, fallo de hardware o ciberataque. Además, se ha cumplido uno de los objetivos del proyecto: que los datos estén protegidos y sean recuperables de forma rápida.

2.5 Servicio Nextcloud

FIGURA 16. Distintas vistas de nextcloud

Introducción

Nextcloud es una herramienta de nube privada que permite a los usuarios guardar, compartir y colaborar con archivos desde cualquier lugar. En este proyecto se ha utilizado como alternativa a servicios como Google Drive o Dropbox, pero con la ventaja de que todo está alojado dentro de la infraestructura de la empresa.

Es uno de los servicios más importantes del sistema, ya que permite que los empleados trabajen con documentos de forma segura, con control de accesos y almacenamiento local.

Instalación en Ubuntu server

Nextcloud no se ha instalado en contenedor Docker si no que se ha realizado la instalación manual sobre Ubuntu Server 22.04 LTS.

Para ello necesitamos instalar Apache, MaraDB y los paquetes necesarios de PHP.

En la base de datos se deshabilito el inicio de sesión remoto para el usuario root y se crearon dos usuarios, nextcloud@localhost y administrador@localhost.

También fue necesario la creación del archivo de configuración virtual para apache en /etc/apache2/sites-available/nextcloud.conf el cual contenía el siguiente texto:

```
<VirtualHost *:80>
        ServerAdmin admin@ipvcs.local
        DocumentRoot /var/www/nextcloud/
        ServerName 10.20.10.12
        ServerAlias www.10.20.10.12
        Alias /nextcloud "/var/www/nextcloud/"

        <Directory /var/www/nextcloud/>
                Options +FollowSymlinks
                AllowOverride All
                Require all granted
                <IfModule mod_dav.c>
                    Dav off
               </IfModule>
                SetEnv HOME /var/www/nextcloud
                SetEnv HTTP_HOME /var/www/nextcloud
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/nextcloud_error.log
        CustomLog ${APACHE_LOG_DIR}/nextcloud_access.log combined
<VirtualHost >
```

TABLA 7. Configuración nextcloud

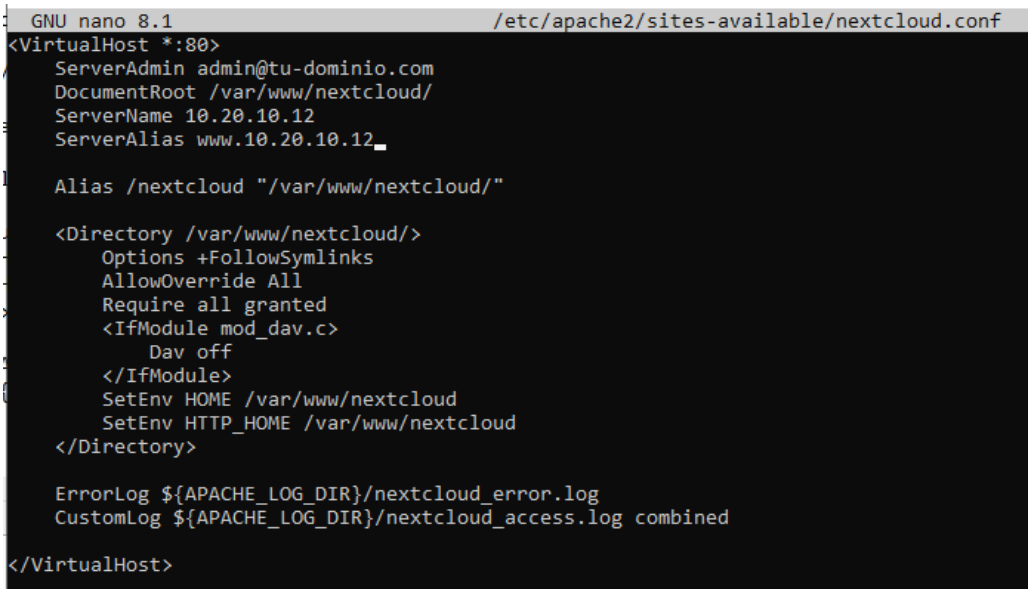


FIGURA 17. Configuración nextcloud

Configuración y usuarios

Una vez instalado, se accedió a la interfaz de Nextcloud y se completó la configuración inicial. Se crearon varios usuarios de prueba, simulando distintos trabajadores de la empresa, y se organizaron por carpetas compartidas según el departamento (marketing, administración, etc.).

También se configuraron permisos para que solo ciertos usuarios pudieran acceder a determinadas carpetas, simulando así un entorno real de empresa con control de acceso a los documentos.

Acceso desde fuera (VPN y DNS interno)

Para que los empleados puedan acceder a Nextcloud desde fuera de la empresa, se ha configurado una VPN con WireGuard usando el contenedor WG-Easy. Este servicio permite conectarse desde cualquier lugar de forma segura, como si estuvieran dentro de la red interna.

Ventajas frente a servicios en la nube

- El control total de los datos lo tiene la empresa.
- No hay costes por usuario ni por almacenamiento.
- El sistema es personalizable y ampliable con plugins.
- El acceso es más rápido desde la red local.
- La información se almacena y se respalda en la misma infraestructura, lo que mejora la protección frente a fallos.

Posibles mejoras

Aunque el sistema ya es funcional, hay varias mejoras que se podrían aplicar si el proyecto se ampliara:

- Integración con LDAP para que los usuarios usen sus cuentas del dominio.
- Activar sincronización de escritorio con los clientes oficiales de Nextcloud.
- Configurar 2FA (doble factor de autenticación) para más seguridad.
- Añadir plugins como calendario, contactos, chat interno, etc.

2.6 Servicio Plex

FIGURA 18. Distintas vistas de plex

Introducción

Plex es una plataforma de gestión y reproducción de contenido multimedia. En este proyecto se ha utilizado para organizar y visualizar vídeos corporativos, tutoriales y otro material relacionado con el departamento de marketing de la empresa.

Se eligió Plex por su compatibilidad con distintos dispositivos, facilidad de uso y su entorno gráfico intuitivo.

Instalación en Ubuntu server

Plex no se ha instalado mediante contenedor Docker. Se ha realizado una instalación manual sobre el servidor srv-docker, siguiendo los pasos oficiales para sistemas basados en Debian/Ubuntu.

Se instaló el paquete apt-transport-https, se añadió el repositorio de Plex a la lista de fuentes de APT, y se instaló Plex Media Server.

Para poder utilizar Plex, es necesario registrarse en su página oficial antes. Se utilizó una cuenta creada con el correo corporativo del centro.

Organización del contenido multimedia

Se creó una carpeta compartida mediante Samba accesible desde la red local con el fin de centralizar todo el contenido que se reproducirá desde Plex.

```
[Recursos compartidos]
  path = /srv/shared
  browsable = yes
  writable = yes
  guest ok = yes
  guest only = yes
  create mask = 0666
  directory mask = 0777
```

FIGURA 19. Imagen de configuración de los recursos compartidos

Una vez montado el entorno, se subieron vídeos simulados para representar el material audiovisual que usaría una empresa. Desde la interfaz de Plex se configuraron las bibliotecas correspondientes para que accediera al contenido de /srv/shared.

Acceso desde la red local

El servicio Plex está disponible desde cualquier dispositivo dentro de la red local mediante la dirección http://10.20.10.12:32400/web.

Esto permite el acceso desde navegadores, televisores, teléfonos móviles o cualquier dispositivo compatible, sin necesidad de configurar acceso externo.

Acceso remoto por VPN

Para acceder a Plex desde fuera de la empresa, se utiliza la VPN creada con WireGuard. Una vez que el usuario se conecta a la red mediante VPN, puede acceder a Plex igual que si estuviera en la oficina.

Esto permite que empleados del departamento de marketing o dirección puedan acceder al contenido multimedia de forma segura desde casa o en movilidad, sin necesidad de abrir puertos en el router ni exponer el servicio a internet.

2.7 Servicio Pi-hole

Introducción

Pi-hole es una herramienta que actúa como servidor DNS interno con funciones de bloqueo de publicidad y filtrado de dominios no deseados. Se ha incluido en este proyecto para mejorar la seguridad y la experiencia de navegación de los usuarios.

Además, permite llevar un registro del tráfico DNS y controlar qué dispositivos acceden a qué dominios.

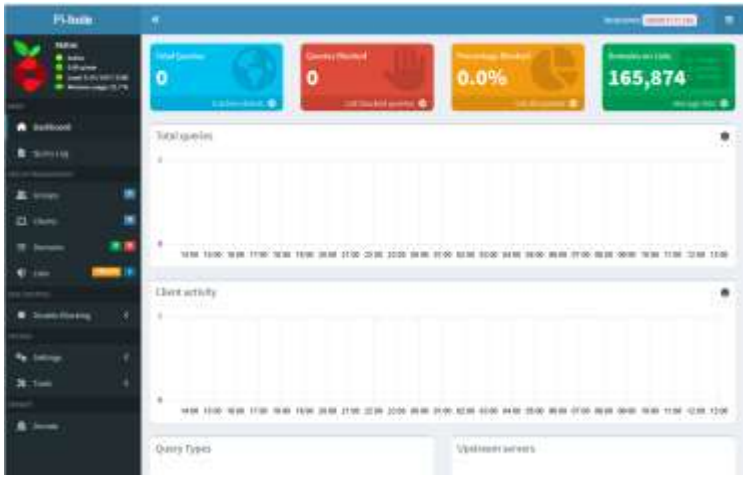


FIGURA 20. Vista principal de pi-hole

Instalación en Docker

Se ha desplegado Pi-hole como contenedor en el servidor Docker, cambiando su puerto de gestión web al 8181 para evitar conflictos con otros servicios. También se ha configurado para usar un volumen persistente, donde guarda la configuración, las listas de bloqueo y el historial.

La configuración se hace a través de una interfaz web accesible desde http://10.20.10.12:9095.

Integración como DNS de la red

pfSense se ha configurado para que Pi-hole actúe como reenviador DNS para todos los dispositivos de la red. De esta forma, todo el tráfico DNS pasa por Pi-hole, permitiendo aplicar filtros centralizados.

FIGURA 21. Muestra de funcionamiento de pihole

2.8 VPN con WireGuard

Introducción

Para permitir el acceso seguro a la red desde el exterior, se ha configurado una VPN con WireGuard, utilizando el contenedor WG-Easy. Esta herramienta facilita la creación y gestión de claves, y ofrece una interfaz web para generar configuraciones de forma rápida.

Gracias a la VPN, los empleados pueden acceder a los servicios internos como Nextcloud o Plex desde casa o cualquier otra ubicación, como si estuvieran conectados en la oficina.

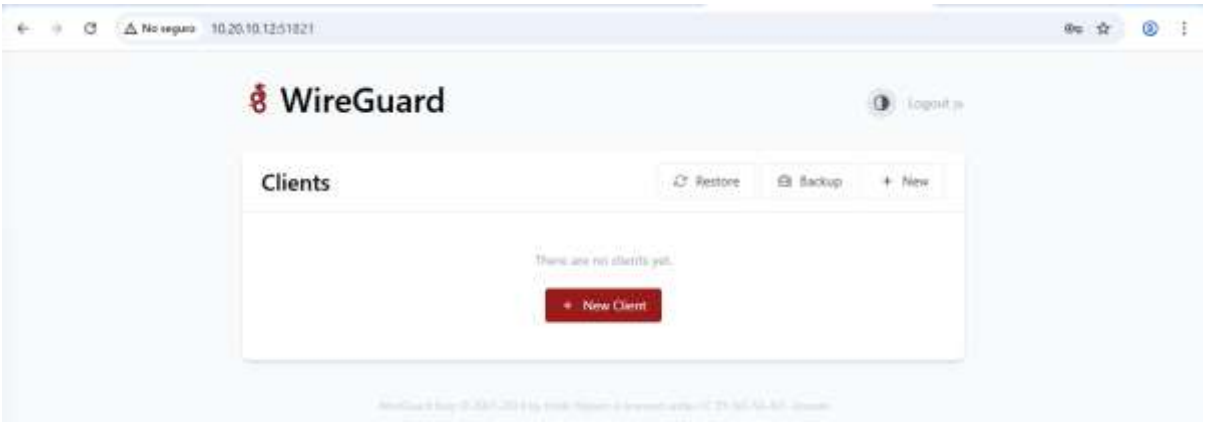


FIGURA 22. Vista de la página principal de wireguard

Configuración del servidor VPN

WG-Easy se ha desplegado en el servidor srv-docker, utilizando el puerto 52828 UDP (personalizado para mayor seguridad). La red de la VPN es 10.98.98.0/24, y cada usuario conectado recibe una IP dentro de ese rango.

La configuración incluye:

- Redirección de tráfico interno.
- DNS interno apuntando a Pi-hole.
- Acceso a todos los servicios internos sin necesidad de abrir puertos públicos.

Se ha generado la clave pública y privada del servidor, y desde ahí se han creado los perfiles para los usuarios que necesiten conectarse.

Además, se ha configurado un dominio dinámico mediante Duck DNS, que permite acceder desde el exterior utilizando un nombre de dominio personalizado en lugar

de depender de la IP pública, que puede cambiar. Esto se ha logrado mediante un script automático que actualiza la dirección IP pública del servidor en Duck DNS. De esta forma, se puede conectar a la VPN desde fuera usando una dirección como midominio.duckdns.org, sin necesidad de realizar cambios manuales cada vez que se reinicia la conexión a internet.

FIGURA 23. Configuración duckdns

Creación de clientes

Desde la interfaz de WG-Easy, se han creado varios perfiles de cliente. Cada uno incluye un archivo .conf y un código QR para facilitar la conexión desde el móvil. Estos perfiles se han guardado en una carpeta compartida a la que solo el administrador tiene acceso.

FIGURA 24. Distintas configuraciones de wg-easy

Una vez importados en la app oficial de WireGuard, el usuario ya puede conectarse a la VPN con un solo clic.

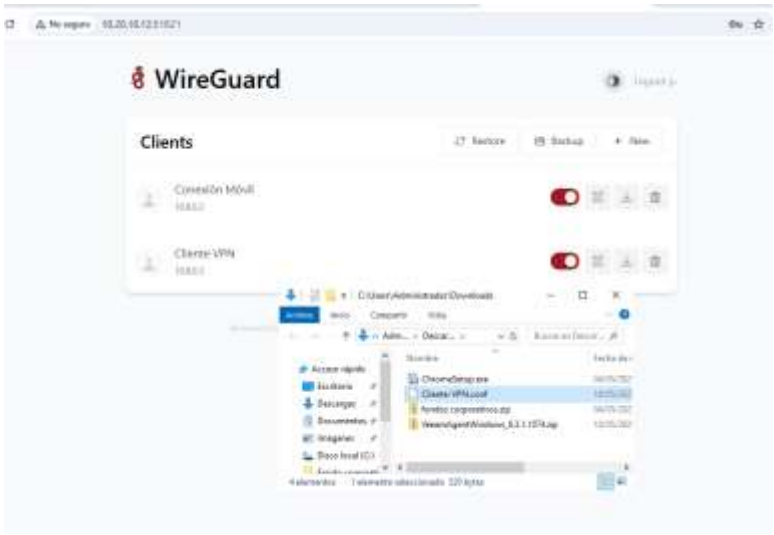


FIGURA 25. Configuración wireguard

Pruebas de conexión y acceso

Se realizaron pruebas de conexión desde fuera de la red simulando un entorno real. Al conectarse con WireGuard, el cliente obtenía una IP dentro de la red de la VPN y podía acceder sin problemas a:

- La interfaz de Nextcloud (http://10.20.10.12/)
- Plex (http://10.20.10.12:32400/)
- La consola de Pi-hole
- Portainer y demás servicios

También se comprobó que podía acceder a recursos compartidos del dominio, haciendo ping y conexiones RDP a los servidores.

Ventajas de usar WireGuard

WireGuard se ha elegido por varias razones:

- Es más rápido que otras VPN como OpenVPN.
- Es muy ligero y fácil de configurar.
- Es compatible con Windows, Linux, macOS y móviles.
- Tiene un alto nivel de seguridad, usando cifrado moderno.

En este proyecto, ha demostrado ser una herramienta muy práctica para acceder a los servicios internos de forma segura y sin complicaciones.

2.9 Monitorización con Zabbix

Introducción

FIGURA 26. Página principal de zabbix

Zabbix es una herramienta de monitorización que permite saber si los servidores y servicios están funcionando correctamente. En este proyecto se ha utilizado para tener control del estado de los servidores srv-docker, srv-ad, y srv-docker2.

Gracias a Zabbix, se pueden detectar fallos antes de que afecten a los usuarios, y también se puede hacer un seguimiento del uso de recursos como CPU, RAM, disco, etc.

Instalación en Docker

Zabbix se ha instalado en contenedores Docker dentro del servidor srv-docker. Se han desplegado los siguientes contenedores:

- zabbix-server-mysql: servidor principal de Zabbix.
- zabbix-web-nginx-mysql: interfaz web.
- mysql-zabbix: base de datos para guardar todos los datos.

También se ha personalizado el puerto de acceso web al 8484 para evitar conflictos. La interfaz está accesible desde zabbix.ipvcsi.local.

Agentes y configuración

En los servidores a monitorizar (Windows y Linux), se ha instalado el agente de Zabbix correspondiente. Este agente envía información del estado del sistema al servidor de Zabbix.

Se han creado los hosts dentro de la interfaz de Zabbix, se ha enlazado cada uno con una plantilla predefinida (Linux Server, Windows Server), y se ha comprobado que los datos llegan correctamente.

Alertas básicas

Se han activado alertas básicas, como:

- Cuando un servidor se apaga.
- Cuando el uso de CPU supera el 80%.
- Cuando el espacio en disco queda por debajo del 20%.

Estas alertas se muestran en la interfaz de Zabbix y permiten actuar rápidamente si algo va mal.

2.10 Copias de seguridad con Duplicati

Introducción

Duplicati es el sistema de copias de seguridad utilizado para proteger los datos de los contenedores Docker. Se ejecuta en un contenedor independiente y realiza copias cifradas de los volúmenes donde se guardan los datos importantes.

Configuración

Desde su interfaz web, accesible desde el puerto 8320, se ha configurado una tarea de copia automática diaria. Esta tarea guarda todos los datos en una carpeta compartida accesible por red desde srv-docker2.

Las copias están cifradas con contraseña y comprimidas para ahorrar espacio. También se ha definido una retención de 7 días, suficiente para este entorno.

Restauración de datos

Se ha hecho una prueba de restauración borrando datos de un contenedor de prueba. Después, desde Duplicati, se seleccionó una copia y se restauraron los archivos en su ubicación original. El resultado fue correcto: los datos se recuperaron tal y como estaban.

Conclusión sobre las copias

Gracias a Duplicati, se garantiza que los servicios del servidor Docker pueden restaurarse fácilmente ante cualquier fallo. Además, la copia está almacenada en otro servidor (srv-docker2), lo que añade un nivel extra de seguridad.

2.11 Monitorización de red con Speedtest

FIGURA 27. Página principal de Speedtest Tracker

Introducción

Para comprobar el estado de la conexión a internet de forma regular, se ha instalado el contenedor de Speedtest en el servidor Docker. Este servicio permite medir la velocidad de subida, bajada y la latencia.

Esto es útil para saber si el proveedor de internet está cumpliendo con lo contratado, o si hay problemas puntuales de red que puedan afectar a los servicios.

Funcionamiento

Speedtest corre en un contenedor accesible desde el puerto 8888. Cada vez que se accede a su interfaz, se puede lanzar un test de velocidad que muestra resultados en tiempo real.

Además, se puede configurar para que se ejecute automáticamente cada cierto tiempo y guarde los resultados, aunque en este proyecto solo se ha utilizado de forma manual para pruebas.

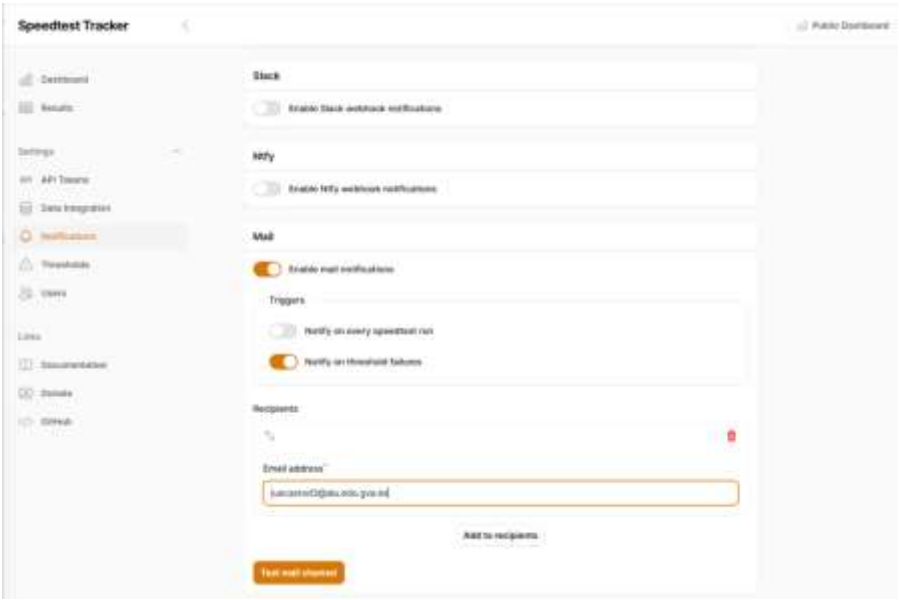


FIGURA 28. Distintas configuraciones de Speedtest Tracker

Resultados y utilidad

Los test realizados durante el desarrollo mostraron una velocidad constante y una latencia baja, lo que indica que la red funciona correctamente.

En un entorno real, esta herramienta puede ser útil para detectar problemas de rendimiento y justificar reclamaciones al proveedor si la velocidad no es la correcta.

2.12 Actualización automática con Watchtower

Introducción

Watchtower es una herramienta que se encarga de mantener actualizados todos los contenedores Docker de forma automática. Funciona como un contenedor más y revisa periódicamente si hay versiones nuevas de los servicios que están instalados.

Configuración y funcionamiento

Watchtower se ejecuta sin interfaz gráfica, pero trabaja en segundo plano. Cada pocas horas analiza los contenedores y, si detecta una nueva versión, detiene el contenedor antiguo, lo actualiza y lo reinicia.

Se ha configurado para que revise las actualizaciones una vez al día. De este modo, el sistema está siempre al día sin intervención manual, lo que mejora la seguridad y el mantenimiento.

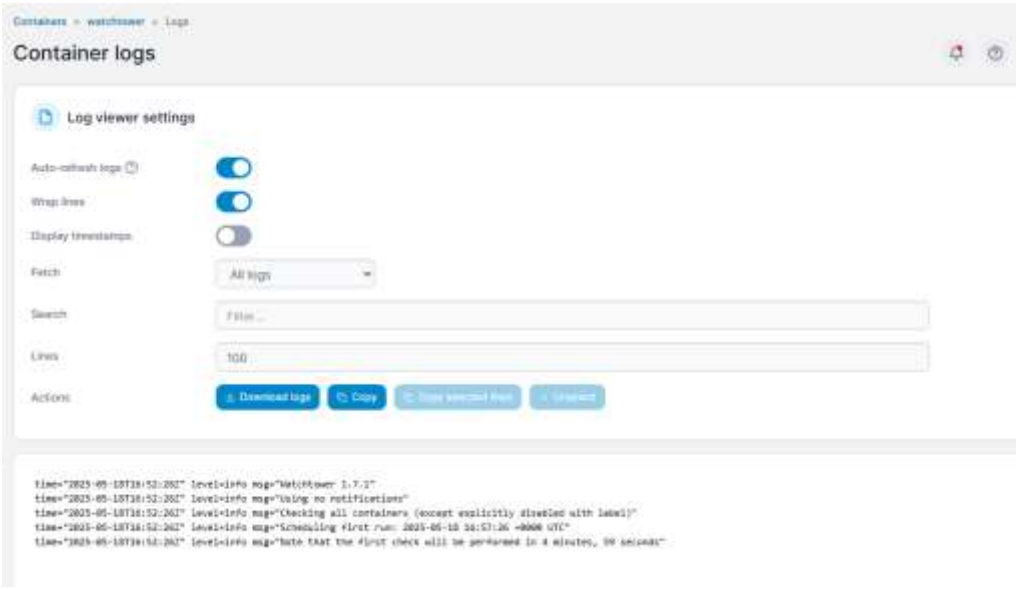


FIGURA 29. Resultados de contenedor watchtower

Beneficios

- Reduce el tiempo de mantenimiento.
- Mejora la seguridad al tener los servicios siempre actualizados.
- Evita errores por olvidos o actualizaciones pendientes.

Es una herramienta especialmente útil en proyectos donde hay muchos contenedores funcionando al mismo tiempo.

3. Mejoras futuras

Durante el desarrollo del proyecto se han planteado varias ideas que no se han podido aplicar por falta de tiempo, pero que se podrían incluir en una segunda fase:

- Integrar Vaultwarden como gestor de contraseñas para usuarios y servicios.
- Conectar Nextcloud y Plex con Active Directory mediante LDAP, para que los usuarios puedan iniciar sesión con sus credenciales del dominio.
- Ampliar el sistema de monitorización con Grafana, para tener paneles visuales en tiempo real.
- Implementar alertas por correo desde Zabbix y Duplicati para avisar de errores o caídas.
- Configurar alta disponibilidad real con balanceadores de carga y clústeres de servicios.
- Desplegar una infraestructura en la nube (como Proxmox o incluso en Azure) para una simulación más cercana al mundo real.

Estas mejoras permitirían dar un paso más en cuanto a seguridad y escalabilidad del sistema.

4. Conclusiones

Este proyecto ha servido para simular la infraestructura de red y servicios de una empresa, aplicando conceptos reales y tecnologías utilizadas actualmente en entornos profesionales.

Se ha conseguido montar una red con dos sedes conectadas por VPN, servidores de dominio con Active Directory, servicios en contenedores Docker, copias de seguridad, monitorización, acceso remoto seguro y una nube privada para compartir archivos.

A lo largo del proyecto se han tomado decisiones técnicas basadas en pruebas reales, ajustando los objetivos al tiempo y recursos disponibles. Se han descartado algunas ideas como Proxmox o Vaultwarden para centrarse en lo esencial, y se han documentado mejoras posibles a aplicar en un futuro.

El resultado es un entorno funcional, seguro y con opciones reales de ampliación si se quisiera llevar a producción o utilizar en un entorno empresarial real. Además, ha permitido aplicar y reforzar conocimientos de redes, sistemas, virtualización, servicios y seguridad de forma práctica.

5. Bibliografía y fuentes

A continuación, se detallan las fuentes de documentación, guías oficiales y materiales de apoyo utilizados durante el desarrollo del proyecto:

- Documentación oficial de pfSense:
  - https://docs.netgate.com/pfsense/en/latest/
  - https://forum.netgate.com/
- Docker Docs: https://docs.docker.com/
- Documentación de Portainer: https://docs.portainer.io/
- WireGuard-easy:
  - https://www.wireguard.com/
  - https://github.com/wg-easy/wg-easy
- Documentación oficial de Nextcloud: https://docs.nextcloud.com/
- Plex: https://gist.github.com/jc-torresp/fa303e1888e93cb51a407273d7636dda
- Duplicati:
  - https://duplicati.readthedocs.io/
  - https://hub.docker.com/r/linuxserver/duplicati
- Veeam Agent for Windows: https://www.veeam.com/
- Zabbix Docs: https://www.zabbix.com/documentation/current/manual/
- Pi-hole Admin Guide:
  - https://docs.pi-hole.net/
  - https://hub.docker.com/r/pihole/pihole
- Speedtest Tracker: https://docs.speedtest-tracker.dev/
- Watchtower: https://tuadmindesistemas.com/instalar-watchtower-con-docker/

6. Anexos

Anexo 1 – Esquema de máquinas virtuales y recursos asignados

Nombre de la VM	Sede	Sistema Operativo	RAM	CPU	Disco (GB)	ISO usada	Función principal
srv-ad	Castellón	Windows Server 2019/2022	4 GB	2	40 GB	ISO oficial de Microsoft	Controlador de dominio, DNS, usuarios
srv-ad2	Madrid	Windows Server 2019/2022	2 GB	2	40 GB	ISO oficial de Microsoft	Controlador de dominio secundario (replica)
srv-docker	Castellón	Ubuntu Server 22.04 LTS	6 GB	2	80 GB	ubuntu-22.04-live-server-amd64.iso	Servicios Docker, contenedores, Ansible
srv-docker2	Madrid	Ubuntu Server 22.04 LTS	4 GB	2	80 GB	ubuntu-22.04-live-server-amd64.iso	Replica de srv-docker, backups
pfSense-Castellón	Castellón	pfSense (FreeBSD)	1 GB	1	8 GB	pfSense-CE-RELEASE.is	Router/firewall, DHCP, VPN, DNS
pfSense-Madrid	Madrid	pfSense (FreeBSD)	1 GB	1	8 GB	pfSense-CE-RELEASE.is	Cliente VPN, firewall, backup de red

TABLA 8. Resumen de máquinas virtuales y recursos

Anexo 2 – docker-compose.yml de cada contenedor del servidor Docker

Wg-easy

```
version: "3.8"
services:
   wg-easy:
      image: ghcr.io/wg-easy/wg-easy
      container_name: wg-easy
      environment:
            - WG_HOST=ipvcs.duckdns.org
            - PASSWORD_HASH=
$$2a$$12$$FnJqGCK9FB9w9xGVU/iTDO8/991dwiKv1u6U6.AZRmScNnkPoohAy
            - WG_DEFAULT_DNS=10.8.1.3 # DNS asignado a los clientes VPN
            - WG_DEFAULT_ADDRESS=10.8.0.x
      volumes:
            - /home/pi/docker/wgeasy:/etc/wireguard
      ports:
            - "51820:51820/udp"
            - "51821:51821/tcp"
      restart: unless-stopped
      cap_add:
            - NET_ADMIN
            - SYS_MODULE
      sysctls:
            - net.ipv4.ip_forward=1
            - net.ipv4.conf.all.src_valid_mark=1
      networks:
            wg-easy_wg-easy:
                  ipv4_address: 10.8.1.2
networks:
      wg-easy_wg-easy:
            external: true
```

Watchtower

```
 version: "3"
services:
     watchtower:
       container_name: watchtower
       image: containrrr/watchtower
       volumes:
             - /var/run/docker.sock:/var/run/docker.sock
       command: --interval 86400 --cleanup
       restart: unless-stopped
```

Pi-hole

```
# More info at https://github.com/pi-hole/docker-pi-hole/ and https://docs.pi-hole.net/
services:
  pihole:
    container_name: pihole-last
    image: pihole/pihole:latest
    ports:
      # DNS Ports
      - "9053:53/tcp"
      - "9053:53/udp"
      # Default HTTP Port
      - "9095:80/tcp"
      # Default HTTPs Port. FTL will generate a self-signed certificate
      - "443:443/tcp"
      # Uncomment the below if using Pi-hole as your DHCP Server
      #- "67:67/udp"
    environment:
      # Set the appropriate timezone for your location (https://en.wikipedia.org/wiki/List_of_tz_database_time_zones), e.g:
      TZ: 'Europe/London'
      # Set a password to access the web interface. Not setting one will result in a random password being assigned
      #FTLCONF_webserver_api_password: 'correct horse battery staple'
    # Volumes store your data between container upgrades
    volumes:
      # For persisting Pi-hole's databases and common configuration file
      - './etc-pihole:/etc/pihole'
      # Uncomment the below if you have custom dnsmasq config files that you want to persist. Not needed for most starting fresh with Pi-hole v6. If you're upgrading from v5 you and have used this directory before, you should keep it enabled for the first v6 container start to allow for a complete migration. It can be removed afterwards
      #- './etc-dnsmasq.d:/etc/dnsmasq.d'
    cap_add:
      # See https://github.com/pi-hole/docker-pi-hole#note-on-capabilities
      # Required if you are using Pi-hole as your DHCP server, else not needed
      - NET_ADMIN
    restart: unless-stopped
```

Duck-dns

```
services:
      duckdns:
            image: lscr.io/linuxserver/duckdns:latest
            container_name: duckdns
            network_mode: host #optional
            environment:
                  - PUID=1000
                  - PGID=1000
                  - TZ=Europe/Madrid
                  - SUBDOMAINS=ipvcs
                  - TOKEN=f688fc9e-287a-46f1-acce-1947786a13d0
                  - UPDATE_IP=ipv4 #optional
                  - LOG_FILE=false #optional
            volumes: - /home/pi/docker/duckdns/config:/config
            restart: unless-stopped
```

Speedtest Tracker

```
version: '3.4'
services:
   speedtest-tracker:
      image: lscr.io/linuxserver/speedtest-tracker:latest
      restart: unless-stopped
      container_name: speedtest-tracker
      ports:
         - 9090:80
         - 8993:443
      environment:
         - PUID=1000
         - PGID=1000
         -
APP_KEY='base64:+sQLBNDpTSJoJLAunWog+gBSu0+aYwGPsuD613LRqpQ='
         - DB_CONNECTION=sqlite
      volumes:
         - /path/to/data:/config
         - /path/to-custom-ssl-keys:/config/keys
```

Anexo 3 – Tabla de accesos para los diferentes servicios

Servicio	Puerto por defecto	Puerto personalizado	Dirección de acceso local
Nextcloud	80/443	8088	http://10.20.10.12
Pi-hole	80/443	9095	http://10.20.10.12:9095/admin
Duplicati	8200	8200	http://10.20.10.12:8200
Speedtest	80/443	9090	http://10.20.10.12:9090
Plex	32400	32400 (sin cambio)	http://10.20.10.12:32400/web
Zabbix	80/443 / 10051	8080 (web) / 10515	http://10.20.10.12/zabbix/
MySQL	3306	3366	(Uso interno, sin acceso web)
WG-Easy VPN	51821/UDP	52828/UDP	Conexión vía cliente WireGuard
DuckDNS	—	—	Sin acceso web – actualiza IP pública
Portainer*	9000 / 9443	9443 (opcional)	http://10.20.10.12:9443

TABLA 1. Servicios, puertos y direcciones de acceso local

Anexo 4 – Información de usuarios y unidades organizativas creadas en AD

Usuario	Nombre completo	OU / Departamento	Función asignada principal
jlopez	Juan López	Marketing	Acceso completo a Plex para subir, organizar y reproducir contenido multimedia.
acastro	Ana Castro	Administración	Acceso a Nextcloud para subir facturas, nóminas y documentación contable.
rruiz	Raúl Ruiz	Soporte IT	Acceso total a todos los servicios, incluido Portainer y Zabbix. Administra copias, VPN y Pi-hole.
cgarcia	Carmen García	Dirección	Acceso a Nextcloud y Plex en modo lectura. Puede consultar informes y vídeos corporativos, pero no modificarlos.

TABLA 1. Usuarios, grupos y funciones que puede realizar cada uno de ellos

Anexo 5 – Procedimiento de emergencia FSMO

Documento anexado en GitHub.

GUÍA COMPLETA DE CONMUTACIÓN Y RECUPERACIÓN DE CONTROLADOR DE DOMINIO (FSMO)

Proyecto IPVCSI – Sedes Castellón y Madrid

Infraestructura del Dominio

Rol	Nombre de equipo	IP	Ubicación	Estado inicial
Controlador de dominio principal (FSMO)	srv-ad1	10.20.10.11	Castellón	Activo
Controlador de dominio de respaldo	srv-ad2	10.30.10.11	Madrid	Replicado

Objetivo del Procedimiento

Simular la caída del controlador de dominio principal (srv-ad1) y garantizar que srv-ad2 asuma la autenticación de usuarios.
Además, si srv-ad1 no va a volver en corto plazo, srv-ad2 deberá asumir los roles FSMO temporalmente.
Cuando srv-ad1 se recupere, deberá recuperar los roles FSMO y continuar como principal.

Escenario 1 – Caída temporal sin transferir FSMO

Este escenario no requiere tomar los roles FSMO, ya que la caída es breve.
Sirve para verificar que srv-ad2 puede mantener el servicio.

1 Apagar srv-ad1

- Desde VirtualBox, apaga srv-ad1 (o desconecta su tarjeta de red).

2 Comprobaciones desde srv-ad2

En srv-ad2, abrir PowerShell y ejecutar:

- Esto verifica que srv-ad2 está autenticando usuarios correctamente.

3 Encender srv-ad1 y verificar replicación

- Inicia srv-ad1 nuevamente.
- Ejecuta en PowerShell:

- Verifica conectividad DNS con:

- Confirma que los roles FSMO siguen en srv-ad1 con:

Escenario 2 – Caída prolongada: Toma forzada de roles FSMO

⚠ Este procedimiento solo debe realizarse si srv-ad1 no va a recuperarse a corto plazo. Se forzará a srv-ad2 a asumir los roles FSMO de forma definitiva.

FASE 1: Transferencia forzada de roles FSMO a srv-ad2

En srv-ad2, abre CMD como administrador:

Esto hace a srv-ad2 a asumir todos los roles FSMO.
Puedes confirmar con:

Escenario 3 – srv-ad1 vuelve y debe recuperar los FSMO

⚠ Asegúrate de que srv-ad1 está sincronizado y replicando correctamente antes de devolverle los roles.

Para ello ejecutamos desde srv-ad1 en PowerShell repadmin /syncall /AdeP

FASE 2: Restaurar FSMO a srv-ad1

En srv-ad1, abre PowerShell como administrador:

- Con esto, srv-ad1 vuelve a ser el titular de los roles FSMO.
- Verifica el resultado con:

Recomendaciones Finales

- Verifica siempre que los relojes están sincronizados con NTP (w32tm /query /status) en ambos DCs.
- Asegúrate de que DNS funciona correctamente entre ambos DCs.
- Usa repadmin /showrepl y dcdiag para diagnósticos adicionales.

Resumen de Comandos Útiles

Comando	Descripción
nltest /dclist:dominio	Ver controladores de dominio disponibles
echo %logonserver%	Ver DC que atiende a un cliente
repadmin /replsummary	Ver estado de replicación global
repadmin /syncall /APeD	Forzar replicación
netdom query fsmo	Ver roles FSMO actuales
w32tm /query /status	Ver estado del servicio horario
whoami /all	Ver tokens de autenticación

Anexo 6 – Tabla de equipos / servicios con su respectivo nombre de usuario y credencial

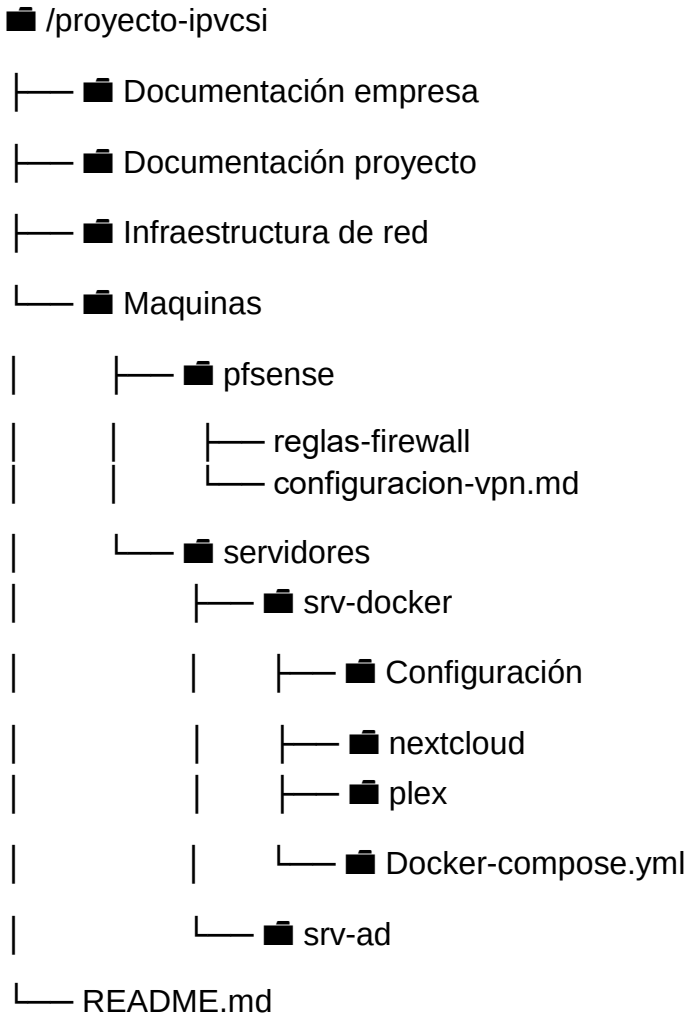
Pagina / Maquina	Usuario	Contraseña
pfsense Castellón	admin	pfsense
pfsense Madrid	admin	pfsense
Srv-ad1	Administrador	Juanjo1234!
Srv-ad2	Administrador	Juanjo1234!
Svr-docker1	administrador	asd
Svr-docker2	administrador	asd
Veeam	Veeam_Admin	Juanjo1234!
Portainer.io	admin	Juanjo24062025!
Speedtest Tracker	Juacasrod3@alu.edu.gva.es	Juanjo1234!
Pi-hole	Admin	Juanjo1234!

TABLA 1. Maquinas /Servicios con su respectivo nombre de usuario y contraseña

7. CONTENIDO DEL REPOSITORIO GITHUB

Estructura del repositorio

El repositorio del proyecto en GitHub contiene toda la documentación técnica, los archivos docker-compose.yml, configuraciones varias, esquemas de red y capturas necesarias para reproducir el entorno. La estructura es la siguiente:



8. RECURSOS UTILIZADOS

Recursos software:

- VirtualBox (gratuito)
- Ubuntu Server y Windows Server (versión educativa)
- pfSense (open source) iso

Licencias: No se han utilizado licencias comerciales fuera del entorno educativo.

9. ÍNDICE DE IMÁGENES

(Generar automáticamente en Word: Referencias > Insertar título > Insertar tabla de ilustraciones)

10. ÍNDICE DE TABLAS