

**Grupo de trabajo 26403GT001:
METODOLOGÍAS COOPERATIVAS EN
AULAS DE INFORMÁTICA**

Memoria

Curso 2025/2026

Dpto. Informática





26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Memoria 26403GT001

Índice de contenidos

1	Introducción.....	4
1.1	Situación de partida.....	4
2	Revisión bibliográfica y propuestas de actuación.....	4
3	Ámbito y planificación de las actuaciones.....	4
3.1	Mejora del cableado y conexiones de las aulas.....	5
3.2	Software de monitorización.....	6
3.2.1	Análisis de soluciones existentes en el mercado.....	6
3.2.2	Descripción de la solución seleccionada: Veyon.....	7
3.2.3	Adecuación a los objetivos del grupo de trabajo.....	7
3.2.4	Ventajas e inconvenientes de la solución.....	8
3.2.5	Conclusión y justificación de la elección.....	8
3.3	Control de comunicaciones.....	8
3.3.1	Ubiquiti UniFi Dream Machine Pro (UDM Pro).....	8
3.3.2	Fortinet FortiGate 60F.....	9
3.3.3	Sophos XGS 87 / XGS 107.....	10
3.3.4	TP-Link Omada ER8411 / ER7206 + Controlador Omada.....	11
3.3.5	pfSense Plus (appliance Netgate 2100 / 4100).....	11
3.3.6	Elección.....	12
3.4	Administración / mantenimiento remoto.....	12
3.4.1	Alternativas.....	13
	Ansible.....	13
	Puppet.....	13
	Chef.....	14
	SaltStack (Salt).....	15
3.4.2	Comparativa.....	16
3.4.3	Análisis específico.....	16
	Criterios ponderados.....	16
3.4.4	Plan de implantación.....	17
4.	Ejecución de actuaciones.....	18
4.1	Mejora del cableado y conexiones de las aulas.....	18
4.2	Software de monitorización.....	19
4.2.1	Situación final.....	25
4.3	Control de comunicaciones.....	25
4.3.1	Configuración del UDM Pro.....	26
4.4	Administración / mantenimiento remoto.....	29
4.4.1	Prueba de concepto.....	29
	Configuración del administrador.....	29
	Configuración de los clientes.....	30
	Preparación de la prueba.....	31



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



4.4.2 Playbooks básicos.....	32
Instalación de software común.....	32
Creación de usuarios.....	33
Uso de IAs en la generación de playbooks.....	34
4.4.3 Despliegue gradual.....	35
4.5 Configuración y maquetado equipo alumnos.....	35
4.5.1 Creación de usuarios.....	36
4.5.2 Supresión de los usuarios.....	36
4.5.3 Creación de la maqueta.....	39
4.5.4 Generalización de la maqueta.....	39
5. Valoración del desempeño.....	39
5.1 Logro de los objetivos del grupo de trabajo.....	40
6. Vías futuras.....	42
7. Autoría y Licencia.....	43



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



1 Introducción

1.1 Situación de partida

El centro dispone de aulas de informática donde el alumnado realiza habitualmente trabajos individuales o por grupos. Sin embargo, la disposición del mobiliario y las características del trabajo con ordenadores dificultan el intercambio de ideas, la comunicación y la colaboración entre iguales, lo que limita el aprendizaje cooperativo.

Asimismo se ha detectado margen de mejora en el uso responsable y educativo de los recursos digitales que el centro pone a disposición del alumnado, lo que impacta negativamente en la dinámica del aula.

El grupo de trabajo ha reflexionado sobre estrategias y recursos que favorezcan el intercambio de conocimientos y experiencias entre el alumnado, potenciando entornos de aprendizaje más participativos y colaborativos buscando minimizar el tiempo que el alumnado dedica a actividades no productivas en el aula.

Como consecuencia de las acciones que acometidas se han incorporado a la configuración de las aulas algunas prácticas y/o tecnologías que facilitarán el trabajo de mantenimiento y administración de las aulas.

2 Revisión bibliográfica y propuestas de actuación

De la revisión de los [artículos](#) propuestos se concluye que resulta interesante el uso de metodologías cooperativas ya que el alumno desarrolla responsabilidades reales, practica competencias profesionales y desarrolla habilidades sociales necesarias en el mundo laboral.

En la puesta en práctica de las metodologías indicadas se detectan riesgos ligados a:

- Bajo rendimiento académico
- Faltas de asistencia reiteradas
- Escasa participación activa en clase

3 Ámbito y planificación de las actuaciones

Tras la revisión documental e identificación de los riesgos se procede a la definición de las líneas de actuación que se focalizan en la mejora de la infraestructura de las instalaciones. Los integrantes del grupo de trabajo deciden estructurar los trabajos en dos fases o grupo de actividades. En la fase I se estudian las alternativas, establecen estrategias y se planifican las líneas de actuación identificadas para ejecutarlas en la fase II.



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



3.1 Mejora del cableado y conexiones de las aulas

La revisión de estado de las comunicaciones en la aulas muestra que la situación es claramente mejorable:

Aula A24:

- No hay canaletas
- No hay rack de comunicaciones
- El switch que hay no es gestionable
- El switch soporta conexiones a 1 gbps
- El cableado está etiquetado y colocado de forma más o menos ordenada.



Figura 1: Switch aula A24

Aula A25:

- El cableado de las mesas del alumnado es adecuado.
- Dispone de canaletas que están en buen estado.
- Es necesario revisar las rosetas de la pared dado que algunas fallan. Faltan tomas, hay más equipos que rosetas funcionales, por lo que hay switches a 10mbps colocados debajo de las mesas para dar servicio a algunos equipos. Este hecho supone una merma en el ancho de banda de esos equipos.
- Existe un rack de comunicaciones que es claramente insuficiente, dado que es muy pequeño y no permite ampliar la instalación existente.
- El cableado del rack requiere de profunda revisión y etiquetado.
- El switch que hay en el rack de comunicaciones es un D-link DES-1024D Switch a velocidad 10/100Mbps.
- El switch no es gestionable y además dadas sus características supone un importante cuello de botella en el rendimiento de la red del aula.



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Figura 2: Switch aula A25

3.2 Software de monitorización

Según se ha identificado por el grupo de trabajo es necesario mejorar los siguientes aspectos:

- Facilitar la supervisión del trabajo del alumnado en tiempo real por parte del profesor/a.
- Favorecer dinámicas de aula más participativas y colaborativas.
- Reducir las incidencias técnicas y el tiempo de gestión de las aulas con ordenadores.
- Garantizar un uso responsable y educativo de los equipos y de Internet (controlar a qué dedica su tiempo cada alumno frente a la pantalla).

Ante esta situación, se considera necesario analizar soluciones de software de control y administración remota que permitan al profesorado gestionar los equipos del aula de forma centralizada, eficiente y segura.

3.2.1 Análisis de soluciones existentes en el mercado

Tras realizar un análisis de mercado, se han identificado diversas herramientas de control y administración remota de equipos informáticos utilizadas en entornos educativos. Entre las más relevantes se encuentran:

- Veyon (software libre y gratuito). - <https://veyon.io/en/>



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



- iTALC (software libre, actualmente menos actualizado). - <https://italc.softonic.com/>
- NetSupport School (software propietario). - <https://www.netsupportschool.com/es/>
- Epointes (entornos Linux). - <https://epointes.org/>
- Microsoft Intune / Endpoint Manager (entornos corporativos y educativos). - <https://adoption.microsoft.com/es-es/microsoft-security/intune/>

De entre ellas, Veyon destaca especialmente por su adecuación al contexto educativo público, al tratarse de una solución gratuita, multiplataforma y específicamente diseñada para aulas TIC.

3.2.2 Descripción de la solución seleccionada: Veyon

Veyon es un software de código abierto que permite al profesorado controlar y administrar de forma remota los equipos del alumnado desde su propio ordenador. Entre sus principales funcionalidades destacan:

- Visualización en tiempo real de las pantallas del alumnado.
- Control remoto de los equipos (ratón y teclado).
- Bloqueo de pantallas y dispositivos.
- Envío de mensajes al alumnado.
- Difusión de la pantalla del profesorado o de un alumno al resto del grupo.
- Apagado, reinicio o cierre de sesión remoto.
- Gestión centralizada sin necesidad de conexión a Internet.

Su instalación es sencilla y puede configurarse para funcionar en red local, lo que favorece la seguridad y la privacidad del entorno educativo.

3.2.3 Adecuación a los objetivos del grupo de trabajo

Relación con los objetivos de formación del profesorado

- Contribuye al desarrollo de la competencia digital docente, al formar al profesorado en herramientas de gestión y control del aula TIC.
- Facilita el aprendizaje de procedimientos básicos de administración y mantenimiento de sistemas informáticos, reduciendo la dependencia técnica externa.
- Permite la elaboración de materiales de referencia y protocolos de uso, transferibles a otros centros.

Relación con los objetivos de impacto en el aula y el centro

- Favorece la presentación de trabajos del alumnado desde sus propios equipos, mejorando la comunicación y el intercambio de ideas.
- Permite un mayor control del uso de los equipos, reduciendo distracciones y accesos no educativos.
- Contribuye a la disminución de incidencias técnicas y del tiempo dedicado al mantenimiento del aula.



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



3.2.4 Ventajas e inconvenientes de la solución

Ventajas

- Software libre y gratuito.
- No requiere conexión a Internet.
- Compatible con sistemas Windows y Linux.
- Diseñado específicamente para entornos educativos.
- Reduce el tiempo de gestión del aula TIC.
- Favorece metodologías activas y colaborativas.

Inconvenientes

- Requiere una configuración inicial por parte del profesorado.
- Necesita formación básica previa para su uso eficiente.
- No incluye filtrado de contenidos web (debe complementarse con otras medidas).

3.2.5 Conclusión y justificación de la elección

Tras el análisis realizado, se considera que Veyon es una solución adecuada y viable para el centro educativo, ya que responde de forma eficaz a las necesidades detectadas en la situación de partida y se ajusta a los objetivos planteados en el grupo de trabajo.

Su implantación permitirá mejorar la gestión de las aulas de informática, optimizar el tiempo del profesorado, fomentar el aprendizaje cooperativo y contribuir al uso responsable y pedagógico de los recursos digitales del centro, avanzando hacia un modelo de uso tecnológico más sostenible e innovador.

3.3 Control de comunicaciones

Con el objetivo de poder gestionar el tráfico generado por las aulas de informática se han seleccionado los siguientes dispositivos:

1. Ubiquiti UniFi Dream Machine Pro (UDM Pro)
2. Fortinet FortiGate 60F
3. Sophos XGS 87 / XGS 107
4. TP-Link Omada ER8411 / ER7206 + Controlador Omada
5. pfSense Plus (appliance Netgate 2100 / 4100)

3.3.1 Ubiquiti UniFi Dream Machine Pro (UDM Pro)

Tipo: Gateway / Firewall / Controlador de red

Precio orientativo: 350 – 420 €

Características principales

- Firewall de nivel profesional
- IDS/IPS (detección y prevención de intrusiones)
- Control de tráfico por dispositivo y por red
- Estadísticas en tiempo real



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



- Filtrado de contenido web
- Gestión centralizada desde interfaz web
- Soporte completo para VLANs
- Integración con switches y puntos de acceso UniFi
- Pensado para instalaciones medianas y grandes

Ventajas

- Interfaz muy intuitiva y visual
- Ideal para centros educativos sin personal técnico dedicado
- Escalable a todo el centro (aulas, WiFi, cámaras)
- Excelente monitorización por aula y por equipo
- Comunidad y documentación muy amplia

Desventajas

- Precio superior a soluciones básicas
- Funcionalidad completa ligada al ecosistema UniFi
- Algunas funciones avanzadas requieren ajuste fino

Curiosidades

- Muy utilizado en universidades y centros de FP
- Permite simular topologías de red reales para aprendizaje
- Se puede administrar remotamente

Conclusión

Es una solución muy equilibrada y completa para nuestro centro educativo, donde lo que buscamos es control, monitorización y crecimiento futuro, siendo ideal como infraestructura central.

3.3.2 Fortinet FortiGate 60F

Tipo: Firewall UTM (Unified Threat Management)

Precio orientativo: 500 – 700 € (sin licencias)

Características principales

- Firewall de alto rendimiento
- Filtrado web avanzado
- Control de aplicaciones
- IDS/IPS profesional
- Monitorización detallada del tráfico
- Soporte VLAN
- Gestión centralizada mediante FortiOS

Ventajas

- Seguridad de nivel empresarial
- Filtrado web muy potente
- Muy estable y fiable



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



- Ampliamente usado en administraciones públicas

Desventajas

- Coste elevado de licencias anuales
- Interfaz menos intuitiva
- Excesivo para aulas pequeñas si no se aprovecha

Curiosidades

- Marca líder en ciberseguridad
- Muy utilizada en hospitales y ayuntamientos
- Ideal para enseñar conceptos de seguridad real

Conclusión

Solución muy robusta, ideal si en nuestro centro priorizásemos la seguridad extrema, aunque puede resultar sobredimensionada para aulas individuales.

3.3.3 Sophos XGS 87 / XGS 107

Tipo: Firewall UTM con control educativo

Precio orientativo: 450 – 650 € (sin licencias)

Características principales

- Firewall y filtrado web avanzado
- Control de aplicaciones
- Protección contra malware
- Monitorización por usuarios y dispositivos
- VLANs y políticas por aula
- Integración con Sophos Central

Ventajas

- Muy buen filtrado de contenidos educativos
- Informes claros y detallados
- Pensado para entornos escolares
- Control granular por grupos

Desventajas

- Requiere licencias para funciones clave
- Curva de aprendizaje media
- Interfaz menos visual que UniFi

Curiosidades

- Muy implantado en colegios privados
- Permite perfiles por edades
- Ideal para control de navegación del alumnado



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Conclusión

Muy buena opción para control de contenidos y uso educativo, especialmente si tuviésemos como máxima prioridad la protección del alumnado.

3.3.4 TP-Link Omada ER8411 / ER7206 + Controlador Omada

Tipo: Gateway / Firewall + Controlador de red

Precio orientativo: 250 – 400 €

Características principales

- Firewall con gestión centralizada
- Control de ancho de banda
- VLANs
- Monitorización de tráfico
- Integración con switches y AP Omada
- Controlador local o en la nube

Ventajas

- Precio muy competitivo
- Gestión centralizada similar a UniFi
- Buena relación calidad/precio
- Adecuado para aulas TIC

Desventajas

- Filtrado web más básico
- Menos opciones de seguridad avanzada
- Comunidad más reducida

Curiosidades

- TP-Link ha crecido mucho en educación
- Ideal para proyectos piloto
- Arquitectura muy similar a UniFi

Conclusión

Alternativa económica y funcional, adecuada si tuviéramos un presupuesto muy limitado y se buscase la gestión centralizada básica.

3.3.5 pfSense Plus (appliance Netgate 2100 / 4100)

Tipo: Firewall software + hardware dedicado

Precio orientativo: 300 – 600 €

Características principales



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



- Firewall de código abierto
- Filtrado web mediante paquetes adicionales
- Monitorización avanzada
- VLANs y reglas complejas
- Altamente configurable
- Muy alto rendimiento

Ventajas

- Extremadamente flexible
- Sin dependencia de marcas cerradas
- Ideal para aprendizaje técnico
- Gran comunidad

Desventajas

- Interfaz menos amigable
- Requiere conocimientos técnicos
- Configuración más lenta

Curiosidades

- Muy usado en FP y universidades
- Perfecto para prácticas reales de redes
- Permite enseñar seguridad avanzada

Conclusión

Excelente opción formativa y técnica, aunque menos adecuada de cara a buscar la simplicidad para el día a día del aula.

3.3.6 Elección

Las características de Ubiquiti UniFi Dream Machine Pro se ajustan muy bien a lo que necesitamos, y su precio es asequible para el Centro. Las posibilidades de control, monitorización y crecimiento futuro son las que consideramos más adecuadas para nuestras aulas de informática, siendo ideal como infraestructura central.

Por ello mismo, y tras sopesar las otras 4 posibilidades, consideramos que la solución que mejor se ajusta a lo que necesitamos es la de **Ubiquiti UniFi Dream Machine Pro**.

3.4 Administración / mantenimiento remoto

Los equipos de los alumnos de Formación Profesional del IES Mar Serena funcionan bajo SSOO Windows y Linux que son mantenidos y actualizados por el profesorado. Las intervenciones sobre los equipos se realizan individualmente en cada uno de ellos. Es por ello que se desea explorar la posibilidad de adoptar alguna herramienta que permita:

- Mantener coherencia en las configuraciones de todos los equipos



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



- Reducir el tiempo dedicado a tareas repetitivas de administración
- Garantizar que las configuraciones y directiva de seguridad se apliquen uniformemente
- Facilitar la implementación de actualizaciones y cambios masivos

3.4.1 Alternativas

Ansible

Descripción general: Ansible es una herramienta de automatización de TI de código abierto que utiliza un modelo sin agentes. Se basa en SSH para Linux y WinRM para Windows, ejecutando tareas definidas en archivos YAML llamados "playbooks".

Arquitectura:

- Modelo push: el servidor de control ejecuta comandos en los nodos gestionados
- Sin agentes: no requiere instalación de software en los equipos cliente
- Comunicación mediante SSH (Linux) o WinRM/PowerShell (Windows)

Ventajas:

- Curva de aprendizaje suave, sintaxis YAML legible
- No requiere agentes en los equipos cliente (menos sobrecarga)
- Documentación extensa y comunidad muy activa
- Módulos específicos para gestión educativa disponibles
- Idempotencia: ejecutar varias veces produce el mismo resultado
- Permite gestionar entornos mixtos Windows/Linux
- Gratuito y open source (Red Hat Ansible Automation Platform es de pago pero la versión community es completa)

Desventajas:

- Requiere Python en los nodos Linux (generalmente ya instalado) y PowerShell en Windows
- Rendimiento puede ser inferior en operaciones masivas comparado con alternativas con agentes
- La gestión de Windows requiere configuración adicional de WinRM
- Para entornos muy grandes puede ser más lento que herramientas con agentes
- Casos de uso en aulas de informática:
 - Instalación y actualización de software en todas las aulas simultáneamente
 - Configuración de usuarios y permisos
 - Despliegue de configuraciones de red
 - Gestión de servicios del sistema
 - Configuración de firewalls locales

Puppet



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Descripción general: Puppet es una de las herramientas de gestión de configuración más maduras. Utiliza un modelo cliente-servidor donde los agentes instalados en cada equipo consultan periódicamente al servidor maestro para obtener su configuración deseada.

Arquitectura:

- Modelo pull: los agentes consultan periódicamente al servidor maestro
- Requiere agentes instalados en todos los nodos
- Lenguaje declarativo propio (Puppet DSL)
- Servidor centralizado (Puppet Master)

Ventajas:

- Muy maduro y estable, utilizado en grandes corporaciones
- Excelente para gestión de configuración a largo plazo
- Reporting detallado del estado de todos los nodos
- Escalable para entornos muy grandes
- Catálogo de módulos extenso (Puppet Forge)
- Versión open source disponible (Puppet Community Edition)

Desventajas:

- Curva de aprendizaje pronunciada, DSL específico no trivial
- Requiere instalación y mantenimiento de agentes en todos los equipos
- Infraestructura más compleja (servidor maestro, base de datos)
- Consume más recursos que alternativas sin agentes
- La versión Enterprise (de pago) tiene características importantes que faltan en la versión gratuita
- Puede ser excesivo habida cuenta del número de equipos que se desean administrar.

Casos de uso en aulas de informática:

- Gestión de configuraciones complejas que dificultan la transferencia de conocimientos a los alumnos
- Entornos donde se necesita reporting detallado y auditoría
- Cuando se planea crecimiento significativo de infraestructura

Chef

Descripción general: Chef es otra herramienta de gestión de configuración empresarial que utiliza Ruby como lenguaje base. Similar a Puppet en filosofía pero con enfoque más programático.

Arquitectura:

- Modelo pull con agentes (Chef Client)
- Servidor centralizado (Chef Server)
- Configuraciones escritas en Ruby (recetas y cookbooks)



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Ventajas:

- Muy potente y flexible para configuraciones complejas
- Lenguaje Ruby permite lógica sofisticada
- Buena integración con cloud y contenedores
- Versionado robusto de configuraciones

Desventajas:

- Curva de aprendizaje muy pronunciada (requiere conocimientos de Ruby)
- Complejidad elevada para entornos pequeños
- Requiere agentes en todos los equipos
- Infraestructura pesada (servidor, base de datos)
- Menos documentación en español que Ansible
- Enfocado principalmente a entornos empresariales grandes

Casos de uso en aulas de informática:

- No se detectan para esta alternativa

SaltStack (Salt)

Descripción general: SaltStack es una herramienta de automatización y gestión de configuración que destaca por su velocidad. Puede funcionar en modo con agentes (Salt Minions) o sin ellos (salt-ssh).

Arquitectura:

- Modelo flexible: puede ser push (salt-ssh) o pull (con minions)
- Comunicación mediante ZeroMQ (muy rápida)
- Configuraciones en YAML con ejecución Python

Ventajas:

- Extremadamente rápido en operaciones masivas
- Flexible: puede funcionar con o sin agentes
- Excelente para ejecución de comandos remotos ad-hoc
- Potente sistema de eventos y orquestación
- Open source con versión enterprise opcional

Desventajas:

- Curva de aprendizaje media-alta
- Documentación menos accesible que Ansible
- Comunidad más pequeña
- Configuración inicial más compleja
- Menos módulos específicos para educación

Casos de uso en aulas de informática:

- Cuando se necesita velocidad de ejecución en operaciones masivas
- Entornos donde se requiere respuesta en tiempo real a eventos



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



- Gestión mixta de servidores y estaciones de trabajo

3.4.2 Comparativa

A continuación se valorarán en una escala de 1 a 5 las distintas alternativas teniendo en cuenta los parámetros que se consideran más interesantes en el entorno del centro educativo, si bien es cierto que no todos ellos deben de tener el mismo peso a la hora de decantarse por alguna alternativa.

Criterio	Ansible	Puppet	Chef	SaltStack
Facilidad de aprendizaje	★★★★★	★★★	★	★★★★
Soporte multiplataforma	★★★★★★	★★★★★	★★★★★	★★★★★
Coste	Gratuito	Gratuito*	Gratuito*	Gratuito*
Recursos necesarios	★★★★★	★★★	★★★	★★★★
Velocidad de ejecución	★★★	★★★★	★★★★	★★★★★★
Comunidad y documentación	★★★★★★	★★★★★	★★★★	★★★★
Escalabilidad	★★★★★	★★★★★★	★★★★★★	★★★★★★
Tiempo de implementación	★★★★★	★★★	★	★★★★

*Versiones enterprise de pago disponibles

3.4.3 Análisis específico

Una vez valoradas las distintas alternativas se procede a graduar las características que se consideran más relevantes en el contexto del centro y alumnado.

- Características prioritarias: soporte multiplataforma y recursos necesarios.
- Características secundarias: escalabilidad, facilidad de aprendizaje y comunidad y documentación.
- Características de bajo impacto: velocidad de ejecución y tiempo de implementación.

Criterios ponderados

Asignando un peso de tres puntos a las características prioritarias, 2 a las secundarias y 1 a las de bajo impacto se obtiene la siguiente tabla:



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Criterio	Ansible	Puppet	Chef	SaltStack
Facilidad de aprendizaje	8	4	2	6
Soporte multiplataforma	15	12	12	12
Recursos necesarios	12	6	6	9
Velocidad de ejecución	3	3	3	5
Comunidad y documentación	10	8	6	6
Escalabilidad	8	10	10	10
Tiempo de implementación	4	2	1	3
Total	60	45	40	51

3.4.4 Plan de implantación

De este estudio se desprende que la opción que más se adecúa a la realidad del centro es la automatización con Ansible. Para llevarla a cabo se define la siguiente secuencia de acciones:

Fase 1: Preparación

1. Instalar Ansible en equipo de administración
2. Configurar acceso SSH a equipos Linux
3. Configurar WinRM en equipos Windows
4. Crear inventario de equipos (archivo hosts)
5. Probar conectividad básica con comandos ad-hoc

Fase 2: Prueba de concepto

1. Crear playbook simple (instalación de paquete, usuario de prueba)
2. Ejecutar en 2-3 equipos de prueba (1 Windows, 2 Linux)
3. Validar resultados
4. Documentar problemas y soluciones

Fase 3: Playbooks básicos

1. Playbook de configuración base Linux
2. Playbook de configuración base Windows
3. Playbook de instalación de software común



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



4. Playbook de creación de usuarios
5. Playbook de configuración de red
6. Uso de IAs en la generación de playbooks

Fase 4: Despliegue gradual

1. Preparar maqueta SO para instalación en PCs de aula.
2. Instalar maqueta en PCs de aula
3. Revisar direccionamiento IP y reajustar ficheros de inventario
4. Extender a segunda aula
5. Documentar procesos

Fase 5: Operaciones

1. Documentación
2. Generación de playbooks a demanda
3. Almacenamiento de ficheros (documentos, configuración y playbooks en repositorio departamental).

4. Ejecución de actuaciones

Una vez establecidas las acciones a realizar e identificadas las dependencias se procede a la ordenación de la ejecución obteniendo así un plan de actuación que se materializa en el siguiente diagrama de Gantt:

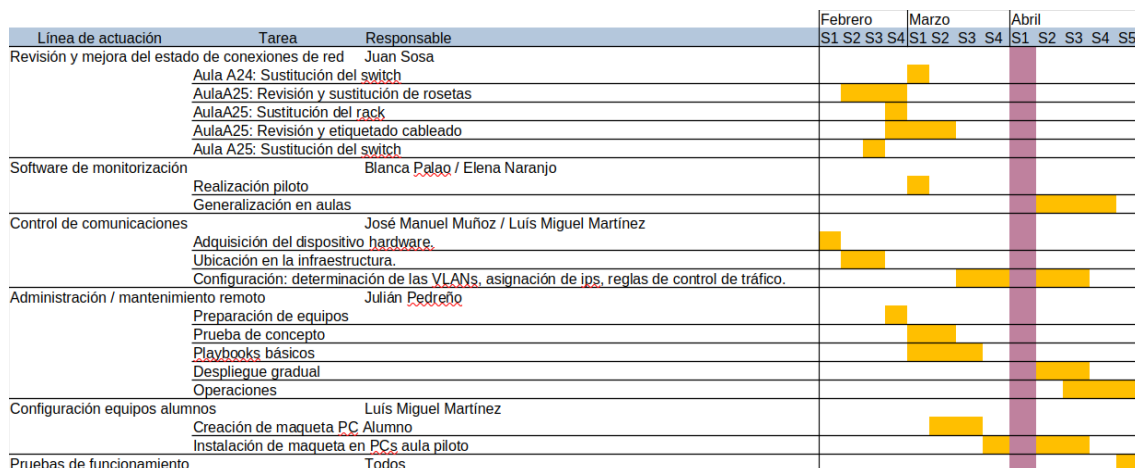


Figura 3: Gantt ejecución fase II

4.1 Mejora del cableado y conexiones de las aulas

Se ha realizado la mejora y actualización del cableado del aula, en concreto las actuaciones que se han llevado a cabo han sido las siguientes:

- Se ha reemplazado el rack de comunicaciones que había por otro más grande. Dado que el armario que había no tenía unas dimensiones adecuadas para la instalación de los dispositivos de red.



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



- Además se ha incorporado al rack un sistema PDU con corte de corriente en caso de derivación, subida o bajada excesiva de tensión, para proteger los equipos activos que hay en el rack de posibles problemas eléctricos.
- Se ha revisado todo el cableado estructurado y se han solucionado problemas de conectividad de algunos equipos.

4.2 Software de monitorización

Se ha optado por usar la autenticación usando claves y la configuración se incorporará a la imagen del sistema que se instalará en los ordenadores de los alumnos. Para implementar poner en funcionamiento el veyon en las aulas de informática se han dado los siguientes pasos:

Ordenador de profesor

1- Instalación del Veyon Configurator y Veyon master y generación de una pareja de claves.

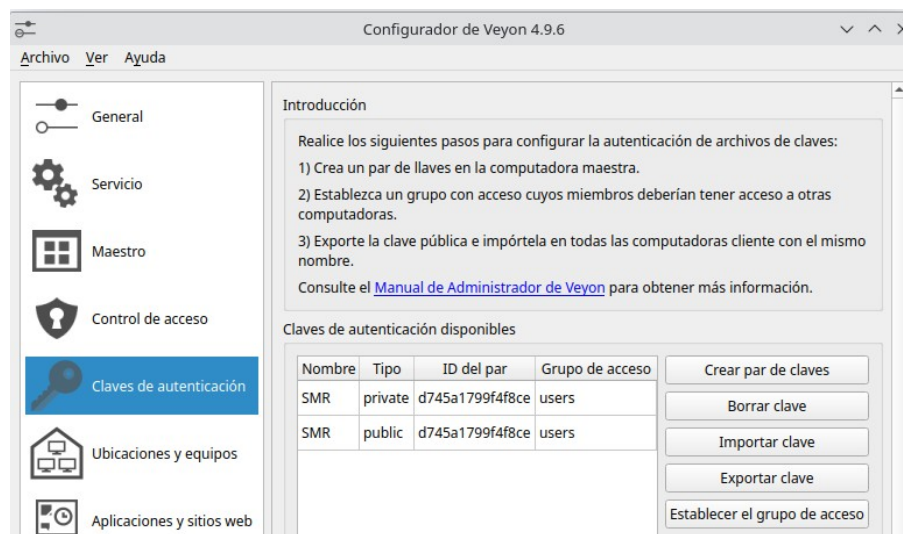


Figura 4: Veyon Configurator - Claves

Haciendo uso de la herramienta de configuración se ha generado una pareja de claves. La clave pública se ha exportado para incorporarla en los ordenadores de los alumnos y la clave privada en los ordenadores de los profesores / administradores del aula. A la hora de importar las claves, éstas deben tener el mismo nombre que el indicado en el administrador ya que en caso contrario no funcionará la autenticación.

2- Definición de los equipos del aula.

Toda vez que se ha adquirido, instalado y configurado el Ubiquiti UniFi Dream Machine Pro se han definido rangos de direcciones (VLANs) para cada una de las aulas y



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



establecido las direcciones IP de los alumnos. Con esta asignación de direcciones se ha procedido a definir las aulas haciendo uso de la herramienta de configuración de Veyon.

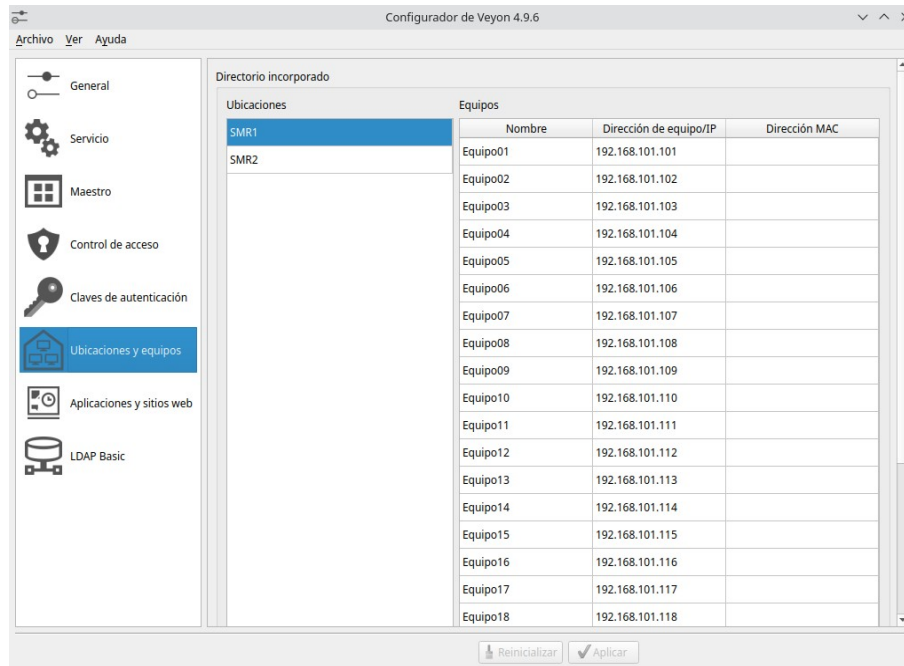


Figura 5: Veyon Configurator - Definición de aulas

Como puede observarse no es necesario informar la MAC.

Ordenador de los alumnos

Los puestos de trabajo de los alumnos deben de tener instalado el servicio de veyon, con la clave pública y configurada la autenticación con clave. Se ha añadido esta configuración en la imagen de los equipos de los alumnos y también se ha desarrollado un playbook de ansible para permitir instalar la herramienta en caso de necesidad.

```
---
- name: Instalar y configurar Veyon en Windows 11
  hosts: windows
  gather_facts: yes
  vars:
    veyon_version: "4.9.6"
    veyon_complete_version: "4.9.6.0"
    veyon_download_url: "https://github.com/veyon/veyon/releases/download/v{{ veyon_version }}/veyon-{{
veyon_complete_version }}-win64-setup.exe"
    veyon_install_path: "C:\\Program Files\\Veyon"
    veyon_service_name: "VeyonService"
    #veyon_master_key_path: "C:\\ProgramData\\Veyon\\keys"
    veyon_master_key_path: "C:\\Program Files\\Veyon\\keys"
    # Ruta local donde se encuentra la clave pública del maestro
    veyon_public_key_source: "files/SMR_public_key.pem"
    # Nombre de la clave
    veyon_key_name: "SMR"
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



```
tasks:
# Verificar que el host es Windows
- name: Verificar sistema operativo Windows
  fail:
    msg: "Este playbook solo es compatible con Windows"
  when: ansible_os_family != "Windows"

# Verificar versión de Windows (Windows 11)
- name: Verificar Windows 11
  debug:
    msg: "Sistema: {{ ansible_distribution }} {{ ansible_distribution_version }}"

# Crear directorio temporal para descargas
- name: Crear directorio temporal
  win_file:
    path: C:\temp\veyon
    state: directory

# Descargar Veyon
- name: Descargar instalador de Veyon
  win_get_url:
    url: "{{ veyon_download_url }}"
    dest: C:\temp\veyon\veyon-setup.exe
    timeout: 600
  register: download_result

- name: Mostrar resultado de descarga
  debug:
    msg: "Descarga completada: {{ download_result.msg }}"

# Verificar si Veyon ya está instalado
- name: Verificar si Veyon está instalado
  win_stat:
    path: "{{ veyon_install_path }}\veyon-master.exe"
  register: veyon_installed

# Instalar Veyon
- name: Instalar Veyon
  win_package:
    path: C:\temp\veyon\veyon-setup.exe
    arguments: '/S /AllUsers /D="{{ veyon_install_path }}"'
    state: present
    wait_for_children: yes
  when: not veyon_installed.stat.exists
  register: install_result

- name: Mostrar resultado de instalación
  debug:
    msg: "Instalación completada"
  when: install_result is changed

# Esperar a que la instalación termine completamente
- name: Esperar a que termine la instalación
  win_wait_for:
    path: "{{ veyon_install_path }}\veyon-master.exe"
    timeout: 120
  when: not veyon_installed.stat.exists

# Crear estructura de directorios para claves
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



```
- name: Crear directorio para claves públicas
win_file:
  path: "{{ veyon_master_key_path }}\public"
  state: directory

- name: Crear directorio para claves privadas (vacío en clientes)
win_file:
  path: "{{ veyon_master_key_path }}\private"
  state: directory

# Copiar clave pública del maestro
- name: Copiar clave pública de Veyon Master
win_copy:
  src: "{{ veyon_public_key_source }}"
  dest: "{{ veyon_master_key_path }}\public\{{ veyon_key_name }}.pem"
register: key_copy_result

- name: Mostrar resultado de copia de clave
debug:
  msg: "Clave pública copiada exitosamente"
when: key_copy_result is changed

# Verificar que la clave se copió correctamente
- name: Verificar clave pública
win_stat:
  path: "{{ veyon_master_key_path }}\public\{{ veyon_key_name }}\key"
register: public_key_check

- name: Confirmar presencia de clave pública
debug:
  msg: "Clave pública encontrada: {{ public_key_check.stat.exists }}"

# Configurar autenticación por claves
- name: Configurar método de autenticación por claves
win_shell: |
  & "{{ veyon_install_path }}\veyon-cli.exe" config set Authentication/Method 1
ignore_errors: yes

# Configurar directorio de claves públicas
- name: Configurar directorio de claves públicas
win_shell: |
  & "{{ veyon_install_path }}\veyon-cli.exe" config set
Authentication/KeyAuthenticationPublicKeyBaseDir "{{ veyon_master_key_path }}\public"
ignore_errors: yes

# Configurar directorio de claves privadas (no usado en clientes)
- name: Configurar directorio de claves privadas
win_shell: |
  & "{{ veyon_install_path }}\veyon-cli.exe" config set
Authentication/KeyAuthenticationPrivateKeyBaseDir "{{ veyon_master_key_path }}\private"
ignore_errors: yes

# Importar la clave pública usando veyon-cli
- name: Importar clave pública en Veyon
win_shell: |
  & "{{ veyon_install_path }}\veyon-cli.exe" authkeys import "{{ veyon_key_name }}"
  "{{ veyon_master_key_path }}\public\{{ veyon_key_name }}.pem"
register: key_import_result
ignore_errors: yes
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



```
- name: Mostrar resultado de importación de clave
debug:
  msg: "Importación de clave: {{ key_import_result.stdout if key_import_result.stdout else
'Completada' }}"

# Configurar el servicio Veyon
- name: Verificar servicio Veyon
  win_service:
    name: "{{ veyon_service_name }}"
  register: veyon_service_check
  ignore_errors: yes

- name: Configurar servicio Veyon para inicio automático
  win_service:
    name: "{{ veyon_service_name }}"
    start_mode: auto
    state: started
  when: veyon_service_check.exists
  ignore_errors: yes

# Configurar firewall para Veyon
- name: Permitir Veyon en el firewall (puerto 11100)
  win_firewall_rule:
    name: "Veyon Server"
    localport: 11100
    action: allow
    direction: in
    protocol: tcp
    state: present
    enabled: yes

- name: Permitir Veyon Master en el firewall
  win_firewall_rule:
    name: "Veyon Master"
    program: "{{ veyon_install_path }}\veyon-master.exe"
    action: allow
    direction: in
    state: present
    enabled: yes

- name: Permitir Veyon Service en el firewall
  win_firewall_rule:
    name: "Veyon Service"
    program: "{{ veyon_install_path }}\veyon-service.exe"
    action: allow
    direction: in
    state: present
    enabled: yes

# Configuraciones adicionales de Veyon
- name: Configurar puerto de conexión
  win_shell: |
    & "{{ veyon_install_path }}\veyon-cli.exe" config set Service/Port 11100
  ignore_errors: yes

- name: Habilitar conexiones desde cualquier IP
  win_shell: |
    & "{{ veyon_install_path }}\veyon-cli.exe" config set Service/AllowedIPs ""
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



```
ignore_errors: yes

# Crear acceso directo en el escritorio
- name: Crear acceso directo de Veyon Master en el escritorio
  win_shortcut:
    src: "{{ veyon_install_path }}\veyon-master.exe"
    dest: C:\Users\Public\Desktop\Veyon Master.lnk
    description: "Veyon Master - Control de escritorios remotos"
    icon: "{{ veyon_install_path }}\veyon-master.exe,0"

# Verificar instalación
- name: Verificar versión instalada de Veyon
  win_shell: |
    & "{{ veyon_install_path }}\veyon-cli.exe" --version
  register: veyon_version_check
  ignore_errors: yes

- name: Mostrar versión de Veyon instalada
  debug:
    msg: "{{ veyon_version_check.stdout }}"
  when: veyon_version_check.stdout is defined

# Verificar estado del servicio
- name: Verificar estado del servicio Veyon
  win_service:
    name: "{{ veyon_service_name }}"
  register: final_service_status
  ignore_errors: yes

- name: Mostrar estado final del servicio
  debug:
    msg: "Estado del servicio Veyon: {{ final_service_status.state if
final_service_status.exists else 'No encontrado' }}"

# Probar conectividad del puerto
- name: Verificar que el puerto 11100 está escuchando
  win_shell: |
    netstat -an | findstr :11100
  register: port_check
  ignore_errors: yes

- name: Mostrar estado del puerto
  debug:
    msg: "Puerto 11100: {{ 'ACTIVO' if port_check.stdout else 'NO ACTIVO' }}"

# Información de configuración
- name: Mostrar información de configuración
  debug:
    msg: |
      Instalación completada:
      - Ruta de instalación: {{ veyon_install_path }}
      - Puerto de servicio: 11100
      - Clave pública configurada: {{ veyon_key_name }}
      - Directorios de claves: {{ veyon_master_key_path }}
      - Método de autenticación: Clave pública
      - Para conectar desde el maestro, usa la clave privada correspondiente

# Limpiar archivos temporales
- name: Limpiar archivos temporales
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



```
win_file:
  path: C:\temp\veyon
  state: absent

handlers:
- name: Reiniciar servicio Veyon
  win_service:
    name: "{{ veyon_service_name }}"
    state: restarted
  ignore_errors: yes
```

Notar que el playbook requiere que la clave privada se encuentre en el subdirectorio files (variable veyon_public_key_source).

4.2.1 Situación final

Una vez configuradas las aulas y distribuidas las claves, tenemos la herramienta operativa. Como consecuencia de ello, ya podemos utilizar la potencia de éste programa en las aulas.

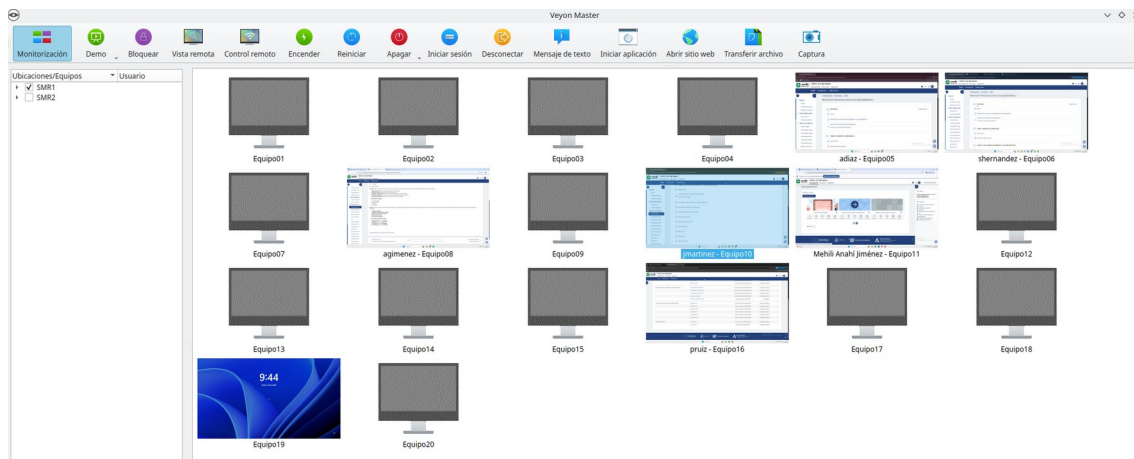


Figura 6: Veyon en funcionamiento

4.3 Control de comunicaciones

Tras la adquisición del Ubiquiti UniFi Dream Machine Pro (UDM Pro) se ha procedido a la ubicación del dispositivo en la infraestructura departamental:



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Figura 7: Enrackado UniFi Dream Machine

4.3.1 Configuración del UDM Pro

Para la configuración del dispositivo se ha hecho uso de la interfaz web proporcionada por el fabricante realizando las siguientes acciones:

Definición de las subredes:

Nombre	ID De ...	Router	Subred	Subred IPv6	DHCP	Concesiones IP	Disponible
● Default	1	INF01	192.168.1.0/24	-	Servidor	8 / 249	241
● SMR1	2	INF01	192.168.101.0/24	-	Servidor	4 / 105	101
● SMR2	3	INF01	192.168.102.0/24	-	Servidor	9 / 249	240
● Dpto	4	INF01	192.168.2.0/24	-	Servidor	1 / 249	248
● A30	5	INF01	192.168.4.0/24	-	Servidor	30 / 249	219

[Crear Nuevo](#) | [Gestionar](#)

Figura 8: Subredes con UDM Pro



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



En las redes de las aulas SMR1 y SMR2 se ha configurado el servidor DHCP para reservar las 150 primeras direcciones para asignación estática. Las direcciones estáticas se usan para poder identificar a los equipos en Veyon.

Nombre
SMR1

Router
INF01

Zona ⓘ
Interno

IPv4

Escalar Red Automáticamente

Dirección IPv4: 192.168.101.1 Máscara de Red: 24

IP del Gateway ⓘ: 192.168.101.1
IP de Difusión ⓘ: 192.168.101.255
IPs Utilizables: 253
Rango de IP: 192.168.101.1 - 192.168.101.254
Máscara de Subred: 255.255.255.0
Tamaño del Pool DHCP: 105
Rango DHCP: 192.168.101.150 - 192.168.101.254

ID de VLAN
2

Figura 9: Configuración subred

Asignación de las subredes a los puertos del dispositivo:

Con la estructura de subredes definida, solo queda indicar la UDM Pro en cual de sus puertos se encuentra conectada cada aula;

Puerto	Anomalia	Nombre	Conexión	Operaci...	Velo...	MAC Conectada	IP Conectada	Perfil	VLAN Nativa	Actividad
<input type="checkbox"/> 1 -- INF01	0	1SMR	-- USW-24-G2		GbE	9c:6b:00:73:1e:68	192.168.101.107	-	SMR1	
<input type="checkbox"/> 2 -- INF01	0	2SMR	-		GbE	d8:bb:c1:dc:ea:bb	-	-	SMR2	
<input type="checkbox"/> 3 -- INF01	0	Port 3	-		Auto	48:ea:02:9d:33:e2	-	-	Default	

Figura 10: Mapeo aula - puerto físico

Definición de las zonas:

La definición de zonas es la "estándar" ubicando todas las redes de las aulas en la red interna.



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA

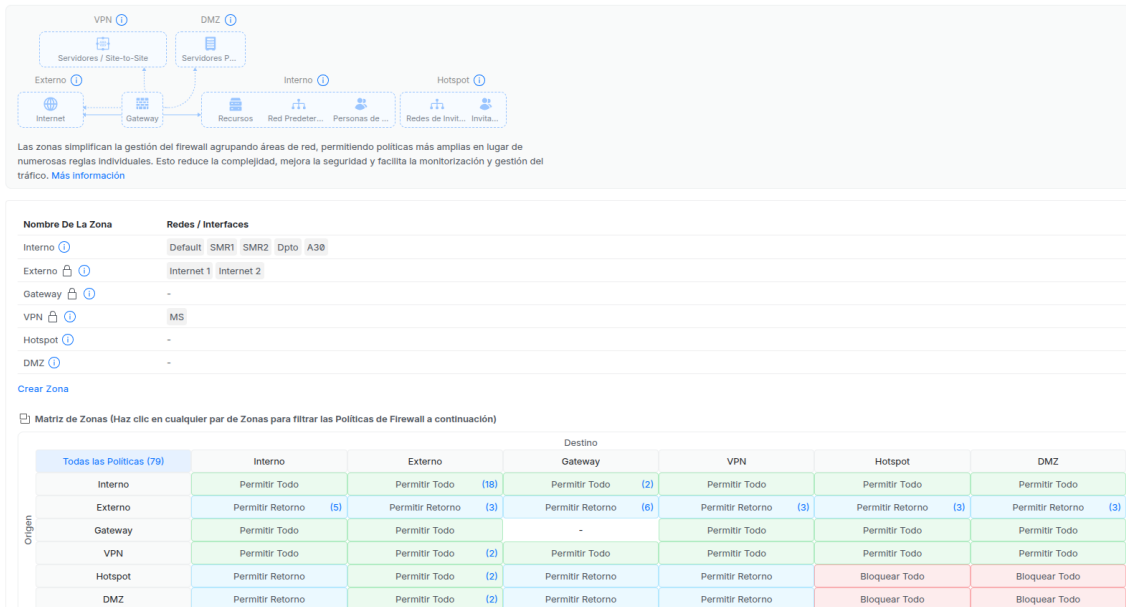


Figura 11: Configuración de seguridad

Definición de las reglas de filtrado:

Para definir una regla es necesario indicar entre otros los siguientes valores:

- Tipo de política: Firewall, Enrutamiento, QoS, NAT, DNS, ACL o Reenvío de puertos.
- Zona de Origen: permite la selección de las redes a las que aplica la regla pudiendo definir reglas distintas según el aula.
- Zona de Destino: para las reglas de filtrado será la red externa y además se debe indicar el tipo de filtrado pudiendo elegir entre filtrado por puertos, por aplicación (trae configurados listados de aplicaciones y grupos de ellas), por dirección IP, por dominio o región.

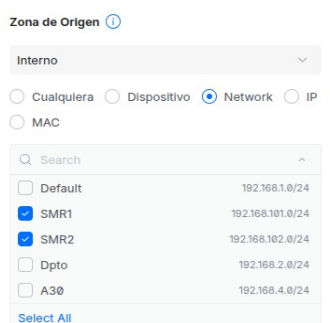


Figura 12: Reglas de filtrado - origen del tráfico

- Acción: Bloquear, permitir o rechazar.



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Nombre	Acción	Versión De IP	Protocolo	Zona Orig.	Orig.	Puerto ...	Zona Dest.	Dest.	Puerto ...	ID	Estado
Aplicaciones bloqu...	Bloquear	Ambos	Todo	Interno	SMR1 SMR2 +1	Cualqui...	Externo	Porn.com PornHub +90	Cualqui...	10006	Todo
Aplicaciones de per...	Bloquear	Ambos	Todo	Interno	SMR1 SMR2 +1	Cualqui...	Externo	Discord	Cualqui...	10004	Todo
Aplicaciones permit...	Permitir	Ambos	Todo	Interno	SMR1 SMR2 +1	Cualqui...	Externo	Spotify FTP Applications	Cualqui...	10001	Todo
Bloquear el tráfico ...	Bloquear	Ambos	Todo	Externo	111.35.151.200	Cualqui...	Interno	Cualquiera	Cualqui...	10001	Todo
Bloquear el tráfico ...	Bloquear	Ambos	Todo	Externo	111.35.151.200	Cualqui...	Interno	192.168.101.120	Cualqui...	10000	Todo
Bloquear el tráfico ...	Bloquear	Ambos	Todo	Interno	192.168.101.120	Cualqui...	Externo	111.35.151.200	Cualqui...	10012	Todo
Bloquear el tráfico ...	Bloquear	Ambos	Todo	Interno	Cualquiera	Cualqui...	Externo	111.35.151.200	Cualqui...	10013	Todo
Dominios bloqueados	Rechazar	Ambos	Todo	Interno	SMR1 SMR2 +1	Cualqui...	Externo	futbin.com crazygames.com +53	Cualqui...	10010	Todo
Dominios de permis...	Permitir	Ambos	Todo	Interno	SMR2 SMR1 +1	Cualqui...	Externo	chatgpt.com claude.ai +1	Cualqui...	10009	Todo
Grupo de aplicacio...	Permitir	Ambos	Todo	Interno	SMR1 SMR2 +1	Cualqui...	Externo	Herramientas de Base de Datos	Cualqui...	10002	Todo
Grupos de aplicacio...	Bloquear	Ambos	Todo	Interno	SMR1 A30	Cualqui...	Externo	Juegos en Línea Redes Sociales +3	Cualqui...	10008	Todo

Figura 13: Reglas de filtrado - Configuración inicial

Es importante notar que la evaluación de reglas de filtrado se hace de forma secuencial y que en caso de que se aplique alguna de ellas se deja de evaluar el resto.

4.4 Administración / mantenimiento remoto

4.4.1 Prueba de concepto

Para la realización de un prueba de funcionamiento de Ansible en nuestro entorno es necesario definir un escenario de prueba que consta de un una máquina linux que hará las veces de administrador y dos clientes (uno linux y otro con windows 11).

Configuración del administrador

El administrador debe instalar ansible:

```
pipx install --include-deps ansible  
pip3 install pywinrm requests
```

```
(base) jpedreno@jpedreno-PCJA:~$ pipx install --include-deps ansible  
Δ Note: ansible-community was already on your PATH at /usr/bin/ansible-community  
Δ Note: ansible was already on your PATH at /usr/bin/ansible  
Δ Note: ansible-config was already on your PATH at /usr/bin/ansible-config  
Δ Note: ansible-console was already on your PATH at /usr/bin/ansible-console  
Δ Note: ansible-doc was already on your PATH at /usr/bin/ansible-doc  
Δ Note: ansible-galaxy was already on your PATH at /usr/bin/ansible-galaxy  
Δ Note: ansible-inventory was already on your PATH at /usr/bin/ansible-inventory  
Δ Note: ansible-playbook was already on your PATH at /usr/bin/ansible-playbook  
Δ Note: ansible-pull was already on your PATH at /usr/bin/ansible-pull  
Δ Note: ansible-test was already on your PATH at /usr/bin/ansible-test  
Δ Note: ansible-vault was already on your PATH at /usr/bin/ansible-vault  
Installed package ansible 13.4.0, installed using Python 3.12.3  
These apps are now globally available  
- ansible  
- ansible-community  
- ansible-config  
- ansible-console  
- ansible-doc  
- ansible-galaxy  
- ansible-inventory  
- ansible-playbook  
- ansible-pull  
- ansible-test  
- ansible-vault  
done! □ □ □  
(base) jpedreno@jpedreno-PCJA:~$ ansible --version  
ansible [core 2.20.3]  
  config file = /etc/ansible/ansible.cfg  
  configured module search path = ['/home/jpedreno/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']  
  ansible python module location = /usr/lib/python3/dist-packages/ansible  
  ansible collection location = /home/jpedreno/.ansible/ansible/collections:/usr/share/ansible/collections  
  executable location = /usr/bin/ansible  
  python version = 3.12.3 (main, Mar 3 2026, 12:15:18) [GCC 13.3.0] (/usr/bin/python3)  
  Jinja version = 3.1.2  
  PyYAML version = 6.0.1 (with libyaml v0.2.5)
```

Figura 14: Instalación Ansible en Linux



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Posteriormente hay que crear los ficheros de inventario en los que se va a indicar los parámetros de conexión y direcciones de los equipos administrados. En esta prueba creamos un único fichero con los datos de conexión a los equipos de pruebas de windows y linux.

```
[win]
192.168.10.160
[win:vars]
ansible_user = AAAAA
ansible_password = BBBBB
ansible_port = 5985
ansible_connection = winrm
ansible_winrm_transport = basic
ansible_winrm_server_cert_validation = ignore
```

En la configuración de windows es necesario indicar que la conexión se realizará usando el servicio de administración remota de windows (winrm) que habrá que activar en el cliente y se ha optado por usar un usuario con permisos de administrador.

```
[linux]
192.168.10.140
[linux:vars]
ansible_ssh_user=root
ansible_ssh_private_key_file=/home/jpedreno/.ssh/id_rsa
ansible_become=true
```

En la prueba de concepto sobre linux se ha optado por testar el acceso utilizando certificados ssh y usar el usuario root directamente.

Generación claves ssh

Generación de la clave ssh en la máquina de administración: `ssh-keygen -t rsa`
Envío de la clave: `ssh-copy-id root@IP cliente linux`

Configuración de los clientes.

Configuración en Windows 11 (válido también para versiones anteriores)

1. Habilitar el servicio **winrm**: `winrm quickconfig`

```
# Activar WinRM
winrm quickconfig -q
# Permitir autenticación básica
winrm set winrm/config/service/auth '@{Basic="true"}'
# Permitir tráfico sin cifrar (solo para intranet)
winrm set winrm/config/service '@{AllowUnencrypted="true"}'
# Abrir firewall
netsh advfirewall firewall add rule name="WinRM HTTP" dir=in action=allow protocol=TCP localport=5985
# Configurar listener
winrm create winrm/config/Listener?Address=**+Transport=HTTP
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



2. Crear un usuario del sistema con privilegios de administrador que se indicará en el fichero de inventario.

Configuración en Linux

No se realizan acciones adicionales toda vez que se ha distribuido la clave ssh.

Preparación de la prueba

Creación del fichero de configuración

En el fichero indicaremos algunas configuraciones tales como la desactivación de la clave ssh del host remoto y forzaremos a ansible a que genere un fichero con los nodos en los que el script ha fallado para que se pueda volver a ejecutar en esos casos.

```
[defaults]
inventory = inventory.ini
host_key_checking = False
retry_files_enabled = True
```

En caso de errores en la ejecución se generará el fichero .retry que se podrá ejecutar con `ansible-playbook playbook.yml --limit @playbook.retry`.

Creación del fichero de inventario

El fichero de inventario contiene los equipos sobre los que se va a ejecutar el playbook junto con las variables en las que indicaremos cómo ha de realizarse la conexión. Para esta prueba definiremos dos grupos [windows] y [linux]

```
[windows]
192.168.101.113

[windows:vars]
ansible_user = XXXX
ansible_password = XXXX
ansible_port = 5985
ansible_connection = winrm
ansible_winrm_transport = basic
ansible_winrm_server_cert_validation = ignore

[linux]
192.168.101.114
[linux:vars]
ansible_ssh_user=root
ansible_ssh_private_key_file=~/.ssh/id_rsa
#para realizar las tareas como root
ansible_become=true
```

Creación del playbook



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Para finalizar crea un fichero que contiene las instrucciones que se ejecutarán en los dos nodos (fichero yml). Para esta primera prueba nos conformaremos con crear un fichero de texto. Es importante notar que las instrucciones del playbook son dependientes del sistema operativo.

```
---
- name: Crear archivo en Linux
  hosts: linux
  tasks:
    - name: Crear archivo de prueba
      file:
        path: /tmp/archivo_ansible.txt
        state: touch

- name: Crear archivo en Windows
  hosts: windows
  tasks:
    - name: Crear archivo de prueba en Windows
      win_file:
        path: C:\Temp\archivo_ansible.txt
        state: touch
```

Lo ejecutaremos con el comando `ansible-playbook playbook.yml -i fichero_inventario`.

4.4.2 Playbooks básicos

Una vez constatada la viabilidad de la alternativa, el siguiente paso consiste en generar algunos playbooks de operaciones que se realizan habitualmente de forma manual en las aulas con el doble propósito de automatizar la tarea y continuar formando al profesorado en la tecnología. Los playbooks se han diseñado para funcionar sobre Windows 11 dado que es el sistema operativo de las aulas objeto de este grupo de trabajo.

Instalación de software común

Para instalar de forma masiva un programa se puede optar por varias opciones: usar la tienda de apps de windows, chocolatey o descargar/distribuir el instalador a las máquinas. En esta primera prueba se va descargar en el directorio temporal de windows el fichero ejecutable que se instalará posteriormente.

```
---
- name: Instalar o actualizar 7-Zip en Windows 11
  hosts: windows
  gather_facts: no

  vars:
    sevenzip_exe_url: "https://www.7-zip.org/a/7z2501-x64.exe"
    sevenzip_exe_path: "C:\\Temp\\7zip.exe"
    sevenzip_install_path: "C:\\Program Files\\7-Zip"
    sevenzip_executable: "C:\\Program Files\\7-Zip\\7z.exe"
  tasks:
    - name: Asegurar que el directorio temporal existe
      ansible.windows.win_file:
        path: C:\Temp
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



```
state: directory

- name: Descargar el instalador EXE oficial de 7-Zip
  ansible.windows.win_get_url:
    url: "{{ sevenzip_exe_url }}"
    dest: "{{ sevenzip_exe_path }}"
    force: yes

- name: Instalar 7-Zip de forma desatendida (EXE)
  ansible.windows.win_command: >
    "{{ sevenzip_exe_path }}" /S
  args:
    creates: "{{ sevenzip_executable }}"

- name: Añadir 7-Zip al PATH del sistema
  ansible.windows.win_environment:
    state: present
    name: PATH
    value: "{{ sevenzip_install_path }}"
    level: machine
```

Creación de usuarios

Otra de las operaciones que al menos se realiza una vez cada curso es la creación de un usuario por equipo para cada alumno. Se opta por restringir los privilegios de éstos haciéndoles formar parte del grupo de invitados. En el playbook se añaden dos variables que contendrán el login y password del usuario y que se podrán especificar en el script o en la llamada al mismo añadiendo el argumento `-e username=nombre_usuario password=pass_usuario`.

```
---
- name: Crear usuarios alumnos en Windows 11
  hosts: windowshost
  gather_facts: yes
  vars:
    username: "nombre_usuario"
    password: "pass_usuario"

  tasks:
    - name: Crear usuario local
      win_user:
        name: "{{ username }}"
        password: "{{ password }}"
        state: present
        groups:
          - Invitados
        password_expired: no
        password_never_expires: yes
        user_cannot_change_password: yes
        account_disabled: no
        description: "Usuario primero de SMR"

    - name: Verificar que el usuario fue creado correctamente
      win_user:
        name: "{{ username }}"
        state: query
      register: user_info
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



```
- name: Mostrar información del usuario creado
debug:
  msg:
    - "Usuario creado: {{ user_info.name }}"
    - "Grupos: {{ user_info.groups }}"
    - "Descripción: {{ user_info.description }}"

- name: Configurar políticas adicionales para el usuario invitado
win_user_right:
  name: "{{ item }}"
  users:
    - "{{ username }}"
  action: remove
loop:
  - SeBatchLogonRight
  - SeServiceLogonRight
  - SeRemoteInteractiveLogonRight
ignore_errors: yes

- name: Permitir inicio de sesión local para el usuario
win_user_right:
  name: SeInteractiveLogonRight
  users:
    - "{{ username }}"
  action: add
```

Este proceso es un poco tedioso ya que obliga a lanzar el script por cada puesto. Si bien es cierto que su ejecución está restringida a un periodo concreto de tiempo, queda como una mejora a realizar en un futuro que el script lea la dirección ip, usuario y password de un fichero de texto u hoja de cálculo.

Uso de IAs en la generación de playbooks

Se ha probado de forma satisfactoria el uso de Chatgpt y Claude para la generación de playbooks usando prompts sencillos. Se recomienda acabar el prompt solicitando a la IA que pregunte lo necesario antes de generar código.



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Necesito un playbook de ansible para borrar un usuario y todo su perfil y ficheros en windows 11. Pregunta lo que necesites antes de generar nada

Antes de generar el playbook, necesito algunos datos clave:
Cuando tengas las respuestas, también me ayudaría saber:

- ¿El usuario puede estar con sesión activa en el momento de ejecutar el playbook? (¿quieres forzar el cierre de sesión antes de borrar?)
- ¿Es un usuario de dominio (AD) o local de la máquina?

Con eso tengo todo lo necesario para generar un playbook limpio y seguro.

P: ¿Cómo se conectará Ansible a los hosts Windows?
R: WinRM (clásico)

P: ¿Qué debe eliminarse exactamente? (Selecciona todas las opciones aplicables)
R: Cuenta de usuario local, Perfil en C:\Users\

P: ¿El usuario a borrar se pasará como variable en tiempo de ejecución o estará hardcodedo en el playbook?
R: Ambas opciones (con valor por defecto)

Figura 15: Empleo de IA en la generación de Playbooks

4.4.3 Despliegue gradual

Para poder extender la funcionalidad a todos los equipos del aula son necesarias dos acciones:

- Establecer una dirección ip fija.
- Incluir la configuración de windows en una imagen (maqueta) para instalar en todos los puestos de las dos aulas.

4.5 Configuración y maquetado equipo alumnos

Para la configuración de los equipos de los alumnos se ha optado por la creación de un equipo piloto con todo el software y configuraciones necesarias, la creación de una imagen del disco y su posterior instalación en los equipos de las aulas.

El inventario de software se ha realizado poniendo en común las necesidades de todos los profesores que actualmente imparten docencia en las aulas identificadas como objeto de este grupo de trabajo. El listado es:

Software

- **SO:** Windows 11 Education
- **Ofimática:** Office 2007 (incluyendo access) y LibreOffice
- **Virtualización:** Oracle Virtual Box
- **Redes:** Cisco Packet Tracer
- **Navegadores:** Chrome, Firefox



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



- **Compresor:** 7zip
- Servicio de Veyon

Configuraciones

- Creación de un usuario administrador para la administración con Ansible
- Configuración del servicio Veyon (certificado y autenticación)
- Activación y configuración del servicio windows RM

4.5.1 Creación de usuarios

El playbook de creación de usuarios se ha realizado como ejercicio en el apartado de [Playbooks Básicos](#).

4.5.2 Supresión de los usuarios

En caso de que un alumno curse su baja y su puesto deba de ser ocupado por otro alumno o haya acabado el curso, se borrarán todas las cuentas de los alumnos junto con los datos almacenados en las carpetas de usuario dejando los equipos dispuestos para ser usados nuevamente. El borrado se efectuará manualmente o bien, haciendo uso del playbook:

```
- name: Eliminar usuario local y todos sus datos en Windows 11
hosts: win
gather_facts: false

vars:
# -----
# Nombre del usuario a eliminar. Puede sobrescribirse con --extra-vars.
# -----
target_user: "olduser"
# Rutas base del perfil (se construyen dinámicamente con target_user)
user_profile_base: 'C:\Users'
# Claves de registro asociadas al perfil de usuario
registry_paths:
- 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList'
- 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles'

tasks:
# -----
# 0. Verificación de seguridad: no permitir borrar cuentas del sistema
# -----
- name: Rechazar si se intenta borrar una cuenta del sistema
  ansible.builtin.fail:
    msg: >
      ERROR DE SEGURIDAD: '{{ target_user }}' es una cuenta del sistema
      protegida. Operación cancelada.
    when: target_user | lower in ['administrator', 'administrador', 'guest','invitado',
'defaultaccount', 'wdagutilityaccount', 'system', 'local service', 'network service']
# -----
# 1. Comprobar si el usuario existe antes de proceder
# -----
- name: Verificar si el usuario '{{ target_user }}' existe en el sistema
  ansible.windows.win_shell: |
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



```
$user = Get-LocalUser -Name "{{ target_user }}" -ErrorAction SilentlyContinue
if ($user) { "EXISTS" } else { "NOT_FOUND" }
register: user_check
changed_when: false

- name: Mostrar advertencia y terminar si el usuario no existe
ansible.builtin.fail:
  msg: "El usuario '{{ target_user }}' no existe en este equipo. No hay nada que eliminar."
when: user_check.stdout | trim == "NOT_FOUND"

- name: Confirmar usuario encontrado
ansible.builtin.debug:
  msg: "Usuario '{{ target_user }}' encontrado. Iniciando proceso de eliminación."
# -----
# # 2. Obtener el SID del usuario (necesario para limpiar el registro)
# -----
- name: Obtener SID del usuario '{{ target_user }}'
ansible.windows.win_shell: |
  (Get-LocalUser -Name "{{ target_user }}").SID.Value
register: user_sid
changed_when: false

- name: Registrar SID obtenido
ansible.builtin.debug:
  msg: "SID del usuario '{{ target_user }}': {{ user_sid.stdout | trim }}"
# -----
# # 3. Eliminar la cuenta de usuario local
# -----
- name: Eliminar la cuenta de usuario local '{{ target_user }}'
ansible.windows.win_user:
  name: "{{ target_user }}"
  state: absent
# -----
# # 4. Eliminar el perfil completo en C:\Users\<usuario>
# Incluye AppData (Local, LocalLow, Roaming) al estar dentro del perfil
# -----
- name: Eliminar directorio de perfil C:\Users\{{ target_user }}
ansible.windows.win_file:
  path: '{{ user_profile_base }}\{{ target_user }}'
  state: absent
# -----
# # 5. Eliminar entradas de registro del perfil (ProfileList)
# La clave del perfil en el registro usa el SID como nombre
# -----
- name: Eliminar clave de registro ProfileList para SID {{ user_sid.stdout | trim }}
ansible.windows.win_regedit:
  path: 'HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\{{ user_sid.stdout |
trim }}'
  state: absent
# -----
# # 6. Eliminar clave de registro HKCU del usuario (rama del hive cargada)
# Si el hive del usuario sigue montado, lo desmontamos y eliminamos
# -----
- name: Desmontar y eliminar hive de registro del usuario si está cargado
ansible.windows.win_shell: |
  $sid = "{{ user_sid.stdout | trim }}"
  $hivePath = "HKU:\$sid"

# Montar la unidad HKU si no existe como PSDrive
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



```
if (-not (Get-PSDrive -Name HKU -ErrorAction SilentlyContinue)) {
    New-PSDrive -Name HKU -PSProvider Registry -Root HKEY_USERS | Out-Null
}
# Descargar el hive si está cargado
if (Test-Path $hivePath) {
    [gc]::Collect()
    Start-Sleep -Seconds 2
    reg unload "HKU\${sid}" 2>$null
}
# Eliminar el fichero NTUSER.DAT si sigue existiendo
$ntuser = "{ user_profile_base } \ { target_user } \ NTUSER.DAT"
if (Test-Path $ntuser) {
    Remove-Item -Path $ntuser -Force -ErrorAction SilentlyContinue
}
"OK"
register: hive_cleanup
changed_when: hive_cleanup.stdout | trim == "OK"
ignore_errors: true
# -----
# 7. Limpiar entradas adicionales de registro relacionadas con el SID
# (NetworkList, CryptProtect, etc.)
# -----
- name: Limpiar entradas adicionales de registro asociadas al SID
  ansible.windows.win_shell: |
    $sid = "{ user_sid.stdout | trim }"
    $keysToDelete = @("HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles",
"HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\LogonUI\SessionData",
"HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU"
)
    foreach ($key in $keysToDelete) {
        $fullKey = "$key\$sid"
        if (Test-Path $fullKey) {
            Remove-Item -Path $fullKey -Recurse -Force -ErrorAction SilentlyContinue
            Write-Output "Eliminado: $fullKey"
        }
    }
# Buscar y eliminar referencias al SID en
# SOFTWARE\Microsoft\Windows\CurrentVersion\Installer
$installerKey = "HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\$sid"
if (Test-Path $installerKey) {
    Remove-Item -Path $installerKey -Recurse -Force -ErrorAction SilentlyContinue
    Write-Output "Eliminado: $installerKey"
}
register: registry_extra_cleanup
changed_when: registry_extra_cleanup.stdout | trim != ""
ignore_errors: true
- name: Mostrar entradas de registro adicionales eliminadas
  ansible.builtin.debug:
    msg: "{ registry_extra_cleanup.stdout_lines }"
  when: registry_extra_cleanup.stdout | trim != ""
# -----
# 8. Resumen final
# -----
- name: Resumen de la operación
  ansible.builtin.debug:
    msg:
      - "=====
      - " ELIMINACIÓN COMPLETADA"
```



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



```
- "===== "  
- " Usuario : {{ target_user }}"  
- " SID : {{ user_sid.stdout | trim }}"  
- " Perfil : {{ user_profile_base }}\\{{ target_user }} → ELIMINADO"  
- " Registro : Entradas del SID → ELIMINADAS"  
- "===== "
```

4.5.3 Creación de la maqueta

Una vez realizadas las instalaciones, configuraciones y comprobada la conectividad y operatividad de la máquina con Veyon y Ansible, se ha procedido a generar una imagen de disco usando clonezilla. Dicha imagen se ha almacenado en varios soportes siendo uno de ellos el NAS departamental accesible a través de samba.

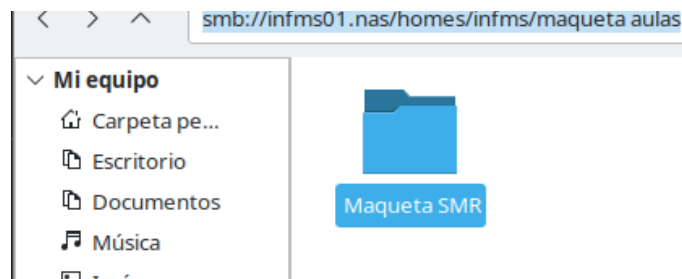


Figura 16: Imagen para puestos de alumnos

4.5.4 Generalización de la maqueta

Con la ayuda del alumnado que está trabajando el RA5 de Montaje y Mantenimiento (Instala software en un equipo informático utilizando una imagen almacenada en un soporte de memoria y justificando el procedimiento a seguir), se ha instalado la imagen del SO en los 40 equipos que forman parte de las dos aulas.

5. Valoración del desempeño

El presente trabajo ha supuesto para el profesorado participante una experiencia enriquecedora ya que hemos ampliado conocimientos técnicos que aplicamos a nuestra actividad docente y hemos involucrado a los alumnos de Formación Profesional en la administración de la infraestructura del centro. También ha resultado satisfactorio el haber contribuido a la mejora de nuestro entorno de trabajo. Adicionalmente se ha producido impacto positivo en el resto de docentes ya que se ha dotado de una serie de herramientas digitales que están aprendiendo a manejar.

La mayor parte de los esfuerzos se han focalizado en la ejecución de tareas de índole técnica, dado que la reflexión inicial puso de manifiesto la necesidad de actualizar nuestras aulas para poder incluir nuevas alternativas metodológicas. Actualmente, la infraestructura se encuentra operativa, aunque que por el momento se carecen de elementos factuales para la evaluación de su impacto en el alumnado.



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



5.1 Logro de los objetivos del grupo de trabajo

En la propuesta del grupo de trabajo se establecieron una serie de de objetivos cuyo nivel de logro pasamos a evaluar a continuación:

Objetivo Formativo	Indicador de Evaluación	Evidencia	Grado de consecución
1. Desarrollar la competencia digital docente del profesorado mediante la formación práctica en herramientas que permitan la interacción y presentación de trabajos del alumnado desde sus propios equipos, favoreciendo un uso didáctico y colaborativo de las TIC	Puesta en marcha al menos una herramienta informática de presentación y control remoto.	Presencia de herramienta presentación y control remoto en las aulas objeto reflejada en la memoria del proyecto.	Se ha instalado, configurado y puesto a disposición del profesorado la herramienta veyon.
2. Adquirir conocimientos técnicos y pedagógicos sobre la gestión segura de redes en entornos educativos, investigando y aplicando estrategias para restringir contenidos no educativos y garantizar un entorno digital responsable y seguro.	Puesta en marcha de al menos un sistema de filtrado y control de contenidos.	Presencia de un sistema de control de tráfico de red en las aulas objeto de actuación reflejado en la memoria del proyecto.	Se ha implantado el control de tráfico de red haciendo uso de firewall Ubiquiti UniFi Dream
3. Formarse en procedimientos de administración y mantenimiento de sistemas informáticos	Establecimiento de normas de configuración de los sistemas para su administración y mantenimiento remoto y centralizado. Empleo de alguna tecnología de administración y mantenimiento remoto.	Guía de configuración de sistemas para su administración y mantenimiento remoto. Utilidades de administración usadas.	Se ha implantado Ansible, generado y ejecutado playbooks para tareas de administración.
4. Diseñar y elaborar materiales de referencia (guía de buenas prácticas, protocolos técnicos o repositorios compartidos) que recojan la experiencia del grupo, favoreciendo la transferencia de conocimiento a otros centros.	Elaboración de memoria del proyecto conteniendo las prácticas puestas en funcionamiento.	Memoria del proyecto.	El presente documento contiene las prácticas y configuraciones realizadas.



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Objetivos de Impacto en Aula o Centro

Objetivos	Indicador de Evaluación	Evidencia Verificable
1. Facilitar que el alumnado pueda presentar sus trabajos desde sus equipos personales o del aula, favoreciendo la comunicación y el intercambio de ideas en entornos tecnológicos colaborativos.	Incorporación de la herramienta de presentación y control en la metodología didáctica	Fotografía

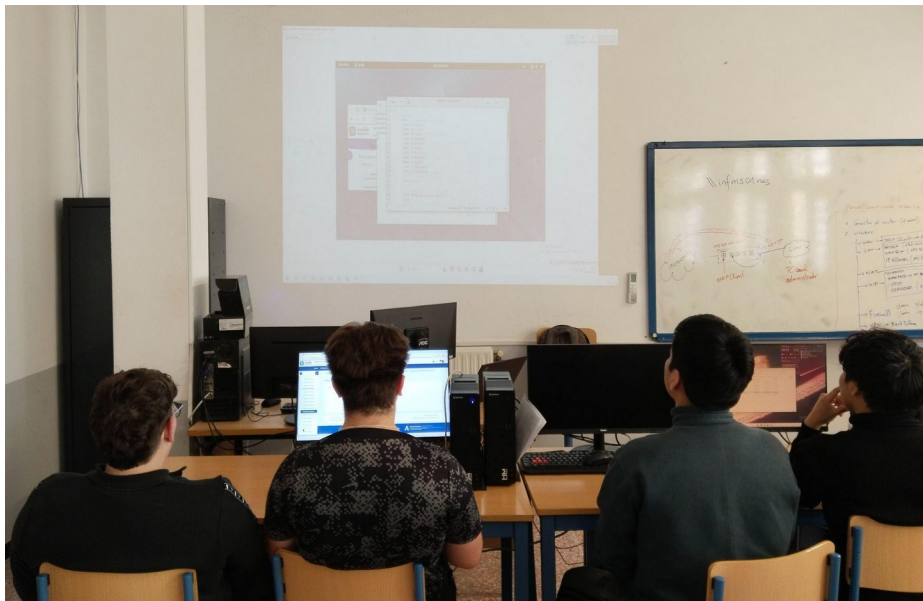


Figura 17: Alumno de 1º SMR explicando su trabajo al grupo

Objetivos	Indicador de Evaluación	Evidencia Verificable
2. Aplicar medidas de control y filtrado de accesos en las aulas TIC, garantizando que las conexiones externas se limiten a contenidos relevantes para el aprendizaje.	Se han filtrado contenidos inapropiados de los equipos del alumno.	Capturas de pantalla desde los equipos de los alumnos



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA

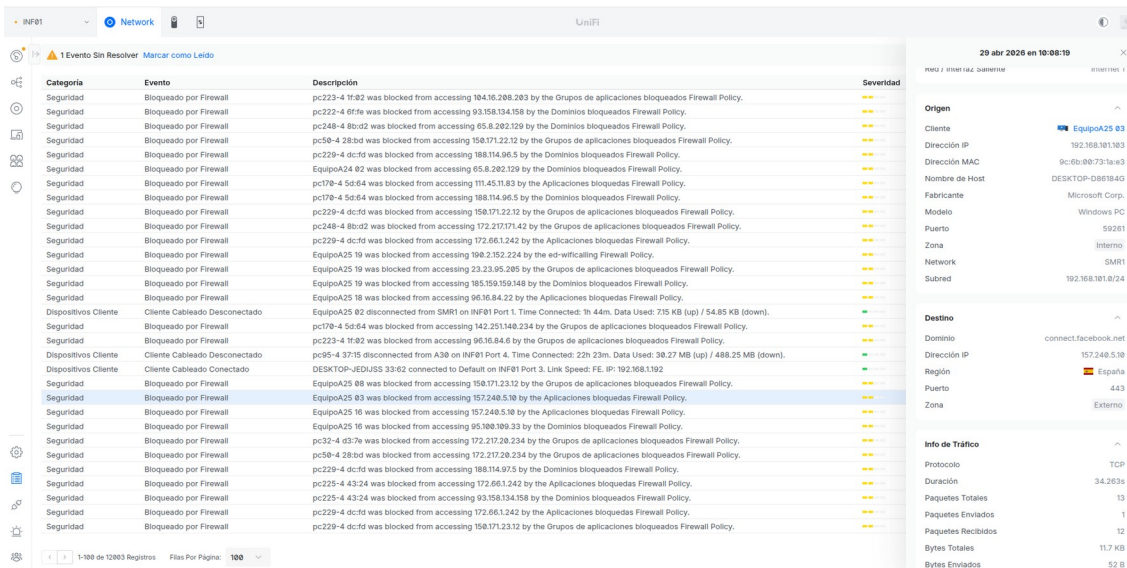


Figura 18: Evidencia del filtrado de tráfico no deseado

Objetivos	Indicador de Evaluación	Evidencia Verificable
3. Reducir las incidencias técnicas y el tiempo de mantenimiento del aula de informática mediante soluciones remotas implementadas y evaluadas por el profesorado participante.	Generación de automatismos para el mantenimiento y administración remota.	Scripts / automatismos generados y presentes en el moodle del grupo.

Los playbooks se encuentran en el moodle del grupo de trabajo y algunos de ellos, a modo de ejemplo en el presente documento.

Objetivos	Indicador de Evaluación	Evidencia Verificable
4. Involucrar al alumnado de Formación Profesional de la familia de Informática y Comunicaciones en actividades prácticas de administración de sistemas, potenciando su aprendizaje aplicado y el trabajo cooperativo con el profesorado	Participación de los alumnos en las actividades de puesta en marcha de las iniciativas.	Fotos durante la realización de las actividades.

6. Vías futuras

En el futuro próximo se debe estabilizar el uso de ambas aulas corrigiendo posibles disfuncionalidades que aparezcan. Con total seguridad habrá que seguir refinando las reglas de filtrado y generando playbooks de Ansible en lugar de realizar configuraciones de forma individual sobre los equipos.



26403GT001 METODOLOGÍAS COOPERATIVAS EN AULAS DE INFORMÁTICA



Para evaluar el impacto de las nuevas medidas se deben establecer los indicadores pertinentes y recoger información. Métricas como el número de actividades cooperativas realizadas, el número de actividades presentadas por los alumnos, el tiempo de dedicado al mantenimiento de equipos o el número de accesos inapropiados bloqueados son algunos indicadores de fácil recolección con lo que construir dichos indicadores.

Si el impacto en la docencia en estas aulas se evalúa positivamente y los indicadores lo refrendan, se podría generalizar la infraestructura a otras aulas lo que implicaría de forma imperativa la formación del profesorado sobre su uso y aprovechamiento.

7. Autoría y Licencia

El presente documento y todos los trabajos reflejados en él han sido acometidos con el apoyo de la asesoría de Dña. Elisabet Iglesias Cano del CEP de Cuevas-Olula por el siguiente grupo de docentes:

- D. Luis Miguel Martínez Portero - Docente de la especialidad de Sistemas y Aplicaciones Informáticas con destino definitivo en el IES Mar Serena.
- D. José Manuel Muñoz Simo - Docente de la especialidad de Informática con destino definitivo en el IES Mar Serena.
- Dña. María Elena Naranjo Suárez - Docente de la especialidad de Sistemas y Aplicaciones Informáticas en régimen de interinidad.
- D. Juan Sosa López - Docente de la especialidad de Informática con destino definitivo en el IES Mar Serena.
- D. Julián Pedreño Gómez - Docente de la especialidad de Informática con destino definitivo en el IES Mar Serena y coordinador del grupo de trabajo.

El presente trabajo está publicado bajo la licencia CC BY-NC-SA 4.0. Para más información visite <https://creativecommons.org/licenses/by-nc-sa/4.0/>