



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Despliegue de imágenes del sistema operativo en aulas departamentales con OpenGnsys y posterior automatización de procesos mediante Powershell

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Federico M. Olmeda de la Casa

Tutores: Juan Carlos Cano Escribá

Juan Luis Posadas Yagüe

Curso 2018-2019

Resum

En este treball s'implementa un sistema de clonació d'imatges base amb el sistema operatiu (Windows 10 Enterprise de 64 bits) junt amb el programari i utilitats que s'utilitzen en la les aules informàtiques d'un departament universitari per a la realització de les pràctiques docents. D'esta manera es disposa d'idèntica configuració en les aules a pesar de tindre equips informàtics amb diferent maquinari permetent la possibilitat d'impartir les pràctiques amb independència de l'aula assignada. Per mitjà de la solució proposada es permet realitzar un manteniment eficient i senzill dels equips informàtics enfront de futures actualitzacions del programari, possibles ampliacions en el nombre d'equips que conformen les aules o fallades de maquinari o programari.

S'ha realitzat també una sèrie de *scripts* emprant la ferramenta de Powershell per a l'automatització de tasques rutinàries que es realitzen ben sovint en l'administració dels equips que conformen les aules informàtiques. D'una banda es facilita la gestió dels sistemes operatius d'una manera prou senzilla i segura i, d'una altra, es minimitzen els possibles errors enfront de la tediosa repetició de tasques si s'optara pel manteniment equip per equip. A més aprofitant que estem en equips amb directori actiu -AD- s'han configurat alguns paràmetres de les directives de grup (GPO) per a complementar i facilitar la configuració del sistema dels equips informàtics.

Paraules clau: Clonació imatges equips, Powershell, *scripting*, GPO.

Resumen

En este trabajo se implementa un sistema de clonación de imágenes base con el sistema operativo (*Windows 10 Enterprise* de 64 bits) junto con el *software* y utilidades que se utilizan en las aulas informáticas de un departamento universitario para la realización de las prácticas docentes. De esta manera se dispone de idéntica configuración en las aulas a pesar de tener equipos informáticos con diferente *hardware* permitiendo la posibilidad de impartir las prácticas con independencia del aula asignada. Mediante la solución propuesta se permite realizar un mantenimiento eficiente y sencillo de los equipos informáticos frente a futuras actualizaciones del *software*, posibles ampliaciones en el número de equipos que conforman las aulas o fallos de *hardware* o *software*.

Se ha realizado también una serie de *scripts* empleando la herramienta de *Powershell* para la automatización de tareas rutinarias que se realizan con frecuencia en la administración de los equipos que conforman las aulas informáticas. Por una parte se facilita la gestión de los sistemas operativos de una manera bastante sencilla y segura y, por otra, se minimizan los posibles errores frente a la tediosa repetición de tareas si se optara por el mantenimiento equipo por equipo. Además aprovechando que estamos en equipos con *directorio activo -AD-* se han configurado algunos parámetros de las directives de grupo (GPO) para complementar y facilitar la configuración del sistema de los equipos informáticos.

Palabras clave: Clonación imágenes equipos, Powershell, *scripting*, GPO.

Abstract

In this work a system of cloning of base images with the operating system is implemented (Windows 10 Enterprise of 64 bits) together with the software and utilities that are used in the computer rooms of a university department for the accomplishment of the teaching practices. In this way the same configuration is available in the classrooms despite having computer equipment with different hardware allowing the possibility of imparting the practices independently of the assigned classroom. Through the proposed solution it is possible to perform an efficient and simple maintenance of the computer equipment against future updates of the software, possible extensions in the number of computers that make up the classrooms or hardware or software failures.

It has also made a series of scripts using the *Powershell* tool for the automation of routine tasks that are frequently performed in the management of the computers that make up the computer rooms. On the one hand it facilitates the management of the operating systems in a fairly simple and safe way and, on the other hand, the possible errors are minimized in the face of the tedious repetition of tasks if one opts for the maintenance team by team. Also, taking advantage of the fact that we are in computers with an active directory -AD-, some parameters of the group policies (GPO) have been configured to complement and facilitate the configuration of the computer equipment system.

Key words: Cloning equipment images, Powershell, *scripting*, GPO.

Índice general

Índice general	V
Índice de figuras	IX
Índice de tablas	XI

Listado de Código	XI
1 Introducción	1
1.1 Antecedentes	1
1.2 Motivación	3
1.3 Objetivos del trabajo	4
1.4 Novedades y contribuciones	4
1.5 Estructura del trabajo	5
2 Herramientas de software para el despliegue de imágenes de un sistema operativo en aulas departamentales	7
2.1 Descripción y características de las herramientas	8
2.1.1 Symantec Ghost Solution Suite	8
2.1.2 Acronis Snap Deploy 5	9
2.1.3 SmartDeploy	10
2.1.4 CloneDeploy	11
2.1.5 Kace SDA	12
2.1.6 Fog Project	13
2.1.7 OpenGnsys	13
2.2 Comparativa de las herramientas anteriores	14
3 Tecnología de clonado	19
4 OpenGnsys	23
4.1 Preparación de la máquina servidora	25
4.1.1 Instalación del sistema operativo	25
4.1.2 Creación de una conexión SSH al servidor OpenGnsys	29
4.1.3 Configuración del segundo disco duro	30
4.2 Instalación de OpenGnsys	32
4.2.1 Versión de ogLive a utilizar	33
4.2.2 Ejecución <i>script</i> instalación	33
4.2.3 Verificación del proceso de instalación	34
4.2.4 Postconfiguración	36
4.2.4.1 Ficheros de configuración de OpenGnsys	37
4.2.5 Instalación de certificado para acceso seguro a la consola Web	37
4.3 Esquema lógico de OpenGnsys	40
4.4 Herramienta de administración de OpenGnsys	42
4.5 Consola de administración	44
4.5.1 Gestión de usuarios	46
4.5.2 Gestión de unidades organizativas	47
4.6 Consola de administración web de una unidad organizativa	49
4.6.1 Herramientas de gestión	50

4.6.2	Gestión del repositorio	51
4.6.3	Organización de aulas	54
4.6.3.1	Gestión de grupos de aulas	55
4.6.3.2	Gestión de grupo de ordenadores	55
4.6.3.3	Gestión de aulas	55
4.6.3.4	Proyecto RemotePC	59
4.6.3.5	Gestión de ordenadores	63
4.6.3.6	Gestión de arranque remoto	64
4.6.4	Acceso SSH a los clientes	66
4.7	Estado de los ordenadores	66
4.7.1	Instalación del agente en el sistema operativo de cliente	67
4.7.2	Ámbito de aplicación y modo de ejecución	68
4.7.3	Operaciones o acciones básicas: los comandos	69
4.7.4	Supervisión y resultado de operaciones	70
4.8	Particionado básico	71
4.9	Crear una imagen del sistema	73
4.9.1	Inventario de hardware del equipo modelo	74
4.9.2	Inventario de software del equipo modelo	75
4.9.3	El objeto imagen monolítica	76
4.9.4	El comando «Crear imagen» de la consola web	77
4.9.5	Formulario de creación de imagen	78
4.10	Despliegue de una imagen del sistema	79
4.10.1	Comando Restaurar	80
4.10.2	Asistente Deploy de imágenes	82
4.10.3	Postconfiguración de los sistemas operativos	84
4.11	Eliminar una imagen del sistema	85
4.11.1	Eliminar la imagen de cache en el cliente	87
4.12	Funcionamiento global de OpenGnsys	88
5	Herramientas para la automatización de procesos en los equipos informáticos	91
6	Introducción al scripting	95
7	PowerShell	97
7.1	¿Qué es PowerShell?	97
7.1.1	PowerShell v.5.1	98
7.1.2	Administración de un equipo en Windows	103
7.1.2.1	Características de Windows	103
7.1.2.2	Aplicaciones de Windows Store	104
7.1.2.3	Los procesos del sistema	104
7.1.2.4	Los servicios del sistema	106
7.1.2.5	El sistema operativo de un equipo	107
7.1.2.6	Protección del sistema	109
7.1.3	Tareas implementadas	110
7.1.3.1	Información del sistema	110
7.1.3.2	Informe de un equipo en HTML	114
7.1.3.3	Obtener un monitor de servidores	124
7.1.3.4	Copia de seguridad	132
7.1.3.5	Limpieza de archivos temporales	136
7.1.4	Otras tareas sencillas pero útiles y necesarias	138
7.1.4.1	Conocer la fecha de encendido de un equipo	138
7.1.4.2	Firma de los scripts mediante certificado	139
7.1.4.3	EventLog	143
7.1.4.4	Máquinas virtuales Hyper-V (VM)	147

7.1.5	Resumen <i>scripts</i> desarrollados	150
8	Conclusiones	153
	Bibliografía	157

Índice de figuras

4.1	Arquitectura de componentes de OpenGnsys	24
4.2	Configuración básica conexión SSH	29
4.3	Consola SSH al equipo donde se va a instalar el servidor OpenGnsys	30
4.4	Comprobación del automontaje del disco de 2 TB	32
4.5	Primeras líneas del <i>script</i> de instalación de OpenGnsys	33
4.6	Estructura de directorios creada con el instalador de OpenGnsys	34
4.7	Verificación de servicios de OpenGnsys	35
4.8	Acceso a la consola web de OpenGnsys	36
4.9	Acceso consola web de OpenGnsys tras su instalación	37
4.10	Modificación configuración Apache para incluir certificado	39
4.11	Acceso seguro a consola web de OpenGnsys	39
4.12	Ejemplo de esquema lógico de una organización con OpenGnsys	41
4.13	Administración descentralizada de una organización con OpenGnsys	42
4.14	Acceso a la vista para el administrador de la consola	42
4.15	Acceso a la vista para el administrador de una OU de OpenGnsys	43
4.16	Área de trabajo de OpenGnsys	44
4.17	Gestión Universidades	45
4.18	Gestión de variables de entorno	45
4.19	Creación usuario superadministrador	46
4.20	Creación usuario administrador	47
4.21	Asignar usuario a una unidad organizativa	47
4.22	Creación de una nueva unidad organizativa	48
4.23	Propiedades sobre una unidad organizativa	48
4.24	Asignación de un usuario administrador sobre una unidad organizativa	49
4.25	Consola de administración web de la unidad organizativa AIM	50
4.26	Barra de herramientas para la gestión de una unidad organizativa	50
4.27	Alta de un nuevo repositorio	52
4.28	Propiedades del repositorio de la OU AIM	53
4.29	Mensaje de error propiedades repositorio	53
4.30	Información del repositorio de la OU AIM	54
4.31	Jerarquía de aulas de la OU AIM	55
4.32	Formulario de propiedades específicas de un aula	56
4.33	Formulario donde se modifican masivamente las propiedades de un aula	56
4.34	Datos de aula física	57
4.35	Datos de servicios de red para el ogLive	58
4.36	Datos del protocolo multicast para un aula	59
4.37	Arquitectura del proyecto RemotePC	60
4.38	Aviso de acceso remoto	60
4.39	Acceso remoto activado	60
4.40	Acceso poliLabs	62
4.41	Acceso identificado a poliLabs	62
4.42	Ventana de lanzar la aplicación UDS	62
4.43	Datos alta de un equipo nuevo	64

4.44	Incorporación de ordenadores a un aula	64
4.45	Gestión arranque avanzado desde OpenGnsys	66
4.46	Estado de los equipos cliente gestionados por OpenGnsys	67
4.47	Modo de ejecución directo	68
4.48	Modo ejecución diferido	69
4.49	Modo ejecución directo y diferido	69
4.50	Modo sin ejecución	69
4.51	Filtro de ejecución de un comando basado en su estado	70
4.52	Cola de acciones sobre el ámbito Aula 2	71
4.53	Información del perfil hardware de un equipo	74
4.54	Información del perfil software de un equipo	75
4.55	Pestaña Imágenes de la consola web OpenGnsys	76
4.56	Propiedades de un objeto de tipo imagen monolítica	77
4.57	Formulario de creación de imagen con OpenGnsys	78
4.58	Concepto de transferencia de datos unicast	79
4.59	Concepto de transferencia de datos multicast	80
4.60	Concepto de transferencia de datos P2P	80
4.61	Comando de restaurar una imagen desde OpenGnsys	81
4.62	Asistente Deploy de imagen desde la consola web de OpenGnsys	82
4.63	Log en tiempo real proceso creación de una imagen desde la consola web de OpenGnsys	83
4.64	Log histórico de un equipo desde la consola web de OpenGnsys	84
4.65	Parte del fichero configureOs donde se ha realizado la postconfiguración de los clientes	85
4.66	Información de una imagen del repositorio que se pretende eliminar	86
4.67	Eliminar un objeto imagen	86
4.68	Eliminar imagen de la caché en cliente desde la consola web de OpenGnsys	88
7.1	Consola PowerShell Cli	99
7.2	Consola PowerShell ISE	100
7.3	Intento de conexión a un equipo remoto	101
7.4	Comandos para habilitar la conexión remota a un equipo	103
7.5	Activación de la característica de Windows 10 de Internet Explorer	103
7.6	Desactivación de la característica de Windows 10 de Internet Explorer	104
7.7	Consulta procesos que ha ejecutado un usuario sobre un equipo remoto	105
7.8	Creación de un nuevo servicio en PowerShell	106
7.9	Listado de los puntos de restauración creados en un equipo mediante PowerShell	109
7.10	Import-Module Get-InformacionEquipo.ps1	112
7.11	Acceso a equipo sin conexión para obtener información del equipo	113
7.12	Acceso a equipo con conexión para obtener información del equipo	113
7.13	Acceso a varios equipo para obtener información de los equipos	113
7.14	Resultado HTML obtenido al ejecutar New-InformeHTML.ps1	123
7.15	Visualización valores Servicios a chequear y tarjetas de red desplegados	123
7.16	Aviso de advertencia informando que el equipo no está disponible	123
7.17	Ejecución <i>script</i> Get-MonitorServidores	128
7.18	Correo electrónico recibido tras ejecutar el <i>script</i> Get-MonitorServidores	128
7.19	Archivo HTML generado por el <i>script</i> Get-MonitorServidores	129
7.20	Prueba ejecución Get-MonitorServidores con un equipo sin conexión	129
7.21	Prueba ejecución Get-MonitorServidores con un equipo con conexión	129
7.22	Prueba ejecución Get-MonitorServidores con varios equipos y uno sin conexión	129

7.23	Añadir usuario a las propiedades de inicio sesión como proceso por lotes	130
7.24	Forzado de actualización de las directivas del equipo	130
7.25	Generación de una tarea programa para monitorizar los servidores del área	132
7.26	Correo electrónico recibido de la función Get-Backup	136
7.27	Ejecución de función "Clean-ArchivosTemporales"	138
7.28	Ejecución de función "Get-FechaEncendidoEquipo"	139
7.29	Listado de políticas de ejecución para la sesión actual	140
7.30	Ejecución de un <i>script</i> sin firmar	140
7.31	Validación del <i>script</i> firmado con un certificado autofirmado	142
7.32	Ejecución de un <i>script</i> firmado	142
7.33	Ejecución de función "Get-InfoSesionesEquipos"	145
7.34	Creación de consultas Get-WinEvent con FilterHashtable	146
7.35	Visualización de Out-GridView para una consulta de eventos en PowerShell	146

Índice de tablas

2.1	Comparativas de las herramientas de la sección 2.1	14
4.1	Instalación Ubuntu Server (Viñetas 1-4) ¹	26
4.2	Instalación Ubuntu Server (Viñetas 5-12) ²	27
4.3	Instalación Ubuntu Server (Viñetas 13-20) ³	28
4.4	Datos de aula física	57
4.5	Información de biarios para los agentes a instalar en los sistemas operativos clientes	68
7.1	Versiones de PowerShell	97
7.2	Propiedades que muestra el cmdlet Get-Process	105
7.3	Descripción de los modos de arranque de los servicios Windows	106
7.4	Políticas de ejecución de PowerShell	140
7.5	Tabla resumen de los <i>scripts</i> que se han desarrollado en este trabajo	151

Listado de Código

7.1	Código Habilitar-PSRemoto.ps1	101
7.2	Get-InformacionEquipo.ps1	110
7.3	Código Get-InfoSO.ps1	114
7.4	Código Get-InfoSistema.ps1	115
7.5	Código Get-InfoDisco.ps1	115
7.6	Código Get-InfoProceso.ps1	116
7.7	Código Get-InfoServicioRevision.ps1	117
7.8	Código Get-InfoRed.ps1	117

7.9	Código New-InformeHTML.ps1	118
7.10	Código Get-MonitorServidores.ps1	124
7.11	Código TareaMonitorServidores.ps1	131
7.12	Código LanzarTareaProgramadaMonitorServidores.ps1	131
7.13	Código Get-Backup.ps1	132
7.14	Código Clean-ArchivosTemporales.ps1	136
7.15	Código Get-FechaEncendidoEquipo.ps1	138
7.16	Código CreacionCertificadoAutofirmado.ps1	141
7.17	Código FirmaScriptFecha.ps1	141
7.18	Código Get-InfoSesionesEquipos.ps1	144
7.19	Código ComandosHyperV.ps1	147
7.20	Código ExportarVM.ps1	148
7.21	Código TareaProgramadaExportarVM.ps1	149

CAPÍTULO 1

Introducción

1.1 Antecedentes

La gestión de los laboratorios informáticos de un departamento universitario ha cambiado mucho en pocos años. Inicialmente se disponían de muy pocos ordenadores para la realización de las prácticas docentes y además sus prestaciones eran bastante reducidas. Se empleaba poco software por la limitación de recursos (el costo del hardware y del software era muy alto). Prácticamente la instalación era equipo por equipo y el ratio de alumnos para cada ordenador era alto.

Pasados unos años se incrementa notablemente el número de ordenadores en las aulas informáticas (se ve reducido notablemente los costos económicos para la adquisición tanto en hardware como en software; además se dispone de aplicaciones de mayor calidad que permite un uso intensivo de los ordenadores en las prácticas docentes universitarias). A la hora de realizar la instalación de las aplicaciones, se hacía en base a las prácticas que se iban a impartir en cada aula debido a que todavía las prestaciones en los equipos y la capacidad de los discos duros no eran excesivas. La tarea que se realizaba no era sencilla porque en la mayoría de las ocasiones los ordenadores que habían en cada aula no eran iguales: se podían encontrar varios modelos de ordenadores con diferentes configuraciones hardware que dificultaban la instalación del sistema operativo y del resto de software necesario. El problema de esta implementación aparecía cuando por problemas técnicos el profesor y los alumnos se tenían que trasladar a otra aula: en la mayoría de las ocasiones no se disponía del software necesario para la impartición de las prácticas y en pocas ocasiones se hacía necesario suspenderlas (lo cual era un problema).

En la actualidad se han incrementado de una manera exponencial las prestaciones de los equipos informáticos y la variedad y calidad de los programas que se emplean. Las empresas desarrolladoras del software son conscientes de la importancia de facilitar sus aplicaciones a las universidades porque en un futuro próximo los alumnos que finalicen sus estudios y empiecen a trabajar en las empresas pueden seguir usando las aplicaciones que aprendieron en la universidad. Ahora se disponen de equipos muy potentes con aplicaciones muy pesadas y complejas. El modo de trabajo actualmente es tener el mismo software instalado en todos los equipos de las aulas del departamento. De esta manera se soluciona el problema comentado anteriormente y, así el alumno puede utilizar las aplicaciones que va a necesitar independientemente del aula donde se encuentre. Como inconvenientes de este método de trabajo es que es necesario discos duros mucho más grandes, procesadores y tarjetas gráficas más potentes y redes cableadas con mayores tasas de transferencia (porque no nos olvidemos que estamos conectados en red y sin ella

nada funcionaría).

Para poder trabajar con un ordenador, primero se ha de instalar un sistema operativo, configurar los drivers de los dispositivos que tenga ese ordenador y aplicar las actualizaciones necesarias del sistema operativo. Posteriormente se necesita instalar las aplicaciones que se vayan a utilizar y, por último, aplicar las configuraciones que sean precisas en el sistema. Cuando se tiene uno o una cantidad reducida de equipos, esta operación se puede realizar uno a uno pero es evidente que cuando se tiene que gestionar un parque de ordenadores de un tamaño considerable, hay que apoyarse en alguna herramienta que nos permita hacer una imagen base de un equipo y distribuirla de alguna forma en el resto de equipos (herramienta de clonado).

La cosa se complica cuando se añaden diferentes requerimientos, como por ejemplo:

- Se dispone de un hardware diferente. No todos los equipos son iguales, ya que se compraron o renovaron en años diferentes.
- La clonación se tiene que realizar utilizando la infraestructura de red que se disponga, por grupos de equipos y, además, debe ser un proceso rápido.
- Se debe poder realizar una configuración de los sistemas tras su instalación (post-configuración).

Por tanto, se necesita una herramienta de clonado para la gestión de los equipos informáticos de las aulas departamentales. Si bien es cierto que hay algunas herramientas en el mercado que reúnen las condiciones indicadas en los puntos anteriores, no hay tantas si se quiere utilizar una herramienta de software libre y que además sea gratuita.

En este TFG, después de valorar varias opciones, se ha elegido OpenGnsys como herramienta de clonado para su utilización en los equipos que conforman nuestro parque informático del Área de Ingeniería Mecánica puesto que satisface todos los requerimientos que se necesitan y presenta una serie de ventajas que se explicarán con más detalle en el capítulo 2.

Para los administradores de sistemas, unos de los puntos débiles de Windows frente a otros sistemas operativos ha sido tradicionalmente la capacidad de administración de este sistema operativo desde la línea de comandos.

Windows siempre ha sido un sistema operativo muy sencillo de administrar visualmente. Sin embargo, esta facilidad de uso, basada en consolas gráficas y asistentes, ha sido un arma de doble filo: por un lado la sencillez con la que se realiza cualquier tarea hace que ciertas personas ganen excesiva confianza pero esta sencillez puede llegarse a convertir en un grave problema que lleva a que los sistemas estén mal administrados debido a que no se ha profundizado bien en los conceptos necesarios para una buena gestión. Por otro lado, también ha sido un problema porque esas mismas interfaces gráficas que facilitan el trabajo más común, se convierten en un estorbo a la hora de realizar tareas complejas que precisen cierto grado de automatización.

En el Área de Ingeniería Mecánica (AIM) se cuenta con equipos que tienen el sistema operativo Microsoft Windows (en sus versiones cliente y servidor). Diariamente se realizan multitud de prácticas docentes y, en muchas de ellas, se utilizan plenamente todos

los equipos disponibles en el aula. Por este motivo, se debe evitar el fallo en algún equipo porque no se puede utilizar hasta que no se haya reparado. Además, hay que tener en cuenta que habitualmente en los departamentos no disponemos de equipos de más para reemplazar los equipos averiados hasta su reparación. Está más que justificado la necesidad de realizar una administración y mantenimiento de los equipos continuo y adecuado no solo para evitar en la medida de la posible el fallo de los equipos sino para tener un rendimiento óptimo de los mismos.

Si se produce un problema en varios equipos a la vez o si es necesario aplicar una nueva función que resuelva un problema en un conjunto numeroso de equipos, no es eficiente que el administrador vaya a cada equipo (uno por uno) y repita siempre los mismos pasos. Por tanto, se necesita una herramienta que permita acelerar procesos y automatizar las tareas rutinarias en gran medida para liberar de trabajo al responsable del sistema.

Se ha decidido realizar la administración del parque de equipos con los que cuenta el AIM con PowerShell. Es una herramienta que cumple perfectamente nuestras necesidades y está totalmente integrada en Windows, gratuita y extremadamente potente. En el capítulo 5 se explica con mayor detalle por qué se ha elegido esta herramienta para la administración de los equipos del AIM. Teniendo clara la herramienta que se va a utilizar para realizar la administración de los equipos del AIM, se van a desarrollar mediante un lenguaje de *scripting* para automatizar procesos repetitivos que se realizan en la administración del sistema.

1.2 Motivación

La motivación de este TFG es la siguiente:

- Facilidad para restablecer un equipo con problemas en el hardware, software o ambos para que esté totalmente operativo en un plazo relativamente corto.
- Mayor control en la supervisión de los equipos de las aulas docentes del área.
- Mejorar el método de trabajo a la hora de restablecer los equipos de las aulas docentes del área. Al menos el proceso de clonación de los equipos se realiza una o dos veces al año, puesto que es muy habitual la política por parte de las empresas que desarrollan el software que se hagan actualizaciones cada seis meses. Además con el cambio de semestre académico se suele utilizar nuevas aplicaciones para la docencia en las aulas que hace necesario el proceso de regeneración de la imagen de las aulas docentes y su posterior clonado a los equipos.
- Mejorar el conocimiento y uso de PowerShell.
- Aplicando la potencia de PowerShell, desarrollar mejores y más sofisticados *scripts* que me permitan una mejor administración de los equipos del AIM.

En definitiva, la motivación de este trabajo es adaptar una herramienta de clonación a la estructura que se dispone en el AIM para disponer siempre operativos los equipos de las aulas (es un proceso que al menos se realiza un par de veces al año) y mejorar el mantenimiento y la administración de los equipos que gestiono en el lugar donde trabajo. Es muy recomendable tener buena actitud e intención de continua mejora para ser mejor

profesional.

La plataforma que se persigue en este trabajo es una plataforma que combina OpenGnsys con PowerShell y *scripts* desarrollados de forma *adhoc* para automatizar tareas de administración del sistema en los equipos del AIM.

1.3 Objetivos del trabajo

Se han establecido los siguientes objetivos para este trabajo académico:

- Implementar mediante OpenGnsys una solución completa para el clonado de imágenes del sistema operativo con las aplicaciones necesarias en los equipos del AIM de la UPV.
- Mejora de conocimientos y uso del *scripting* en PowerShell.
- Implementación de *scripts* desarrollados en PowerShell para automatizar tareas de administración de sistemas.
- Administración de equipos desde la línea de comandos PowerShell.
- Utilización de herramientas gratuitas que se adapten a las necesidades actuales del AIM.

1.4 Novedades y contribuciones

Las novedades que se presentan en este trabajo son:

- La utilización de herramientas gratuitas.
- La utilización de **OpenGnsys** como herramienta de clonado para disponer en todos los equipos de las aulas docentes la misma configuración de sistema operativo y aplicaciones.
 - Se va a instalar y configurar un servidor OpenGnsys desde cero.
 - Se va a configurar los equipos del AIM para poder arrancar por PXE y conectarse al servidor de OpenGnsys.
 - Se va a crear una imagen base con el sistema operativo que se ha utilizado y se ha realizado unas operaciones de actualización tanto del propio sistema operativo como de los drivers del equipo. Como disponemos de varios equipos de aula con configuración distinta, se tendrá que ir actualizando dicha imagen base para que al final se tengan los drivers de cualquier equipo de los que disponemos en el AIM.
 - Con esa primera imagen base con todos los drivers que se puedan necesitar en los equipos de las aulas, se va a instalar en un equipo todas las aplicaciones necesarias para la impartición de docencia y se ha verificará que se ha configurado adecuadamente. Estando seguros del correcto funcionamiento de ese equipo, se va a proceder a crear una nueva imagen del aula con todo (sistema operativo, aplicaciones y configuraciones necesarias).
 - Se va a desplegar la imagen creada anteriormente a todos los equipos de aulas que disponemos.

- Se va a realizar una tarea de postconfiguración en los equipos para tener la configuración necesaria (cambio de nombre del equipo, meter en el dominio, antivirus, etc.).
- La utilización de **PowerShell** como herramienta para realizar el mantenimiento y la administración adecuada de los equipos que hay en el Área de Ingeniería Mecánica (AIM).
 - Se van a automatizar tareas de administración sobre equipos del AIM desde una consola de PowerShell sin movernos de nuestro equipo de trabajo.
 - Se va a realizar la administración remota de equipos del AIM.
 - Se va a recopilar información personalizada de los equipos del AIM.
 - Se va a tratar la seguridad de los *scripts* firmando los archivos con un certificado autofirmado.
 - Se va a gestionar las máquinas virtuales bajo Hyper-V que dispone el AIM mediante PowerShell.
 - Se va a realizar copias de seguridad entre directorio (origen y destino), limpieza de archivos temporales, monitorización personalizada de servidores y acceso al registro de eventos mediante PowerShell.

Con respecto a las contribuciones que he aportado en este trabajo (se centran en los capítulos cuatro y siete) han sido:

- Adaptación de la herramienta OpenGnsys a la infraestructura que se dispone actualmente en el Área de Ingeniería Mecánica.
- Creación de las operaciones que han sido necesarias para poder disponer en todos los equipos de las aulas que tenemos de idéntica configuración (sistema operativo y aplicaciones) para la impartición de docencia.
- Creación de *scripts* para la administración de los equipos del AIM y la automatización (mediante la creación de tareas programadas) de tareas creadas previamente en PowerShell.

1.5 Estructura del trabajo

El resto de este trabajo se ha organizado de la siguiente manera:

En el capítulo 2 se realiza un análisis de herramientas de software para el despliegue de imágenes de un sistema operativo y se justifica la elección de OpenGnsys como solución elegida para este trabajo.

En el capítulo 3 se introduce la tecnología de clonado de equipos informáticos cuyo fin es instalar de una manera sencilla y rápida en varios equipos la misma configuración de programas y sistema operativo de un equipo base.

En el capítulo 4 se presenta la herramienta OpenGnsys que se ha elegido para la implementación del despliegue de imágenes base al resto de equipos que conforman nuestro parque de ordenadores del AIM. Se comienza desde cero realizando la instalación y configuración inicial de esta herramienta, se describe el esquema lógico que utiliza y las

herramientas de administración que posee.

En el capítulo 5 se mencionan algunas herramientas para la automatización de procesos en los equipos y se justifica la elección de PowerShell como solución elegida para este trabajo.

En el capítulo 6 se introduce brevemente el concepto de scripting: qué es y por qué se recomienda su uso.

En el capítulo 7 se presenta PowerShell. A partir de los requisitos y necesidades del sistema, se ha desarrollado los programas en lenguaje de *scripting* mediante PowerShell necesarios para:

- Realizar la administración remota de equipos.
- Gestionar máquinas virtuales bajo Hyper-V.
- Realizar copias de seguridad de un directorio, limpieza de archivos temporales de un equipo, acceso al registro de eventos y monitorización personalizada de servidores.
- Recopilar información personalizada de equipos.

Finalmente, en el capítulo 8 se presentan las conclusiones de este trabajo y se indican algunas líneas de trabajo futuras.

CAPÍTULO 2

Herramientas de software para el despliegue de imágenes de un sistema operativo en aulas departamentales

El presente trabajo está desarrollado en el Área de Ingeniería Mecánica (AIM) del Departamento de Ingeniería Mecánica y de Materiales (DIMM) de la Universitat Politècnica de València. Se dispone de entre quince y veinte ordenadores en cada aula. Para realizar un uso óptimo de las mismas, se mantienen todas con el mismo sistema operativo (Windows 10) y con las mismas aplicaciones instaladas para que se puedan impartir las asignaturas en cualquier aula de las que se dispone. Se imparten clases de diferentes grados y algunas asignaturas de máster para cuyas prácticas se hace necesario la instalación de aplicaciones ingenieriles de mucha complejidad y consumo de espacio en el disco duro. Es por ello que las imágenes que se necesitan son muy pesadas y es necesario disponer de discos duros de un tamaño considerable. A pesar que se dispone de una partición adicional para guardar los datos generados por los usuarios de los equipos de las aulas, los alumnos suelen guardar los datos en sus perfiles. Esto implica un aumento considerable del espacio utilizado del disco del sistema. Por tanto se hace necesario realizar mantenimiento continuo de los equipos para evitar inconvenientes como quedarte sin espacio en el disco del sistema (esto ralentizaría notablemente su rendimiento).

Como el número de equipos a gestionar es relativamente grande se necesita el uso de técnicas de clonación de equipos para facilitar la tarea. En la actualidad todos los equipos de las aulas del AIM utilizan el sistema operativo de Microsoft. Por tanto necesitamos emplear alguna herramienta de software para poder desplegar una imagen base que se haya creado para la impartición de la docencia en los diferentes equipos que se disponen.

Los requisitos que se necesitan para valorar dichas herramientas y que cumpla las necesidades que se tienen en el AIM son:

- Funcionamiento para sistemas Microsoft Windows.
- Facilidad de uso.
- Flexibilidad.
- Coste económico de las licencias de utilización.

- Posibilidad de automatización para que haya poca interacción con el técnico encargado.

Se va a realizar un pequeño análisis de opciones que se disponen en el mercado y que se adapten a las necesidades actuales para el despliegue de imágenes. La necesidad del empleo de herramientas para el despliegue de imágenes en equipos informáticos ha crecido considerablemente y con ello el número de soluciones que las empresas te ofrecen. Es por ello que probablemente no se haya tenido en cuenta todas las opciones que hay actualmente en el mercado.

2.1 Descripción y características de las herramientas

Las herramientas que se van a analizar y comparar son las siguientes:

- Symantec Ghost Solution Suite.
- Acronis Snap Deploy 5.
- SmartDeploy.
- CloneDeploy.
- Kace SDA.
- FOG Project.
- OpenGnsys.

2.1.1. Symantec Ghost Solution Suite

La información mostrada se ha obtenido de la hoja de datos del producto [1].

Symantec Ghost™ Solution Suite es una solución galardonada y líder en la industria para implementar y administrar computadoras de escritorio, computadoras portátiles, tabletas y servidores. Desde una única consola de administración, puede migrar rápida y fácilmente a los últimos sistemas operativos, inventariar máquinas, implementar software y realizar configuraciones personalizadas en múltiples plataformas de hardware y tipos de sistemas operativos, incluidos Windows, Mac y Linux.

Ghost Solution Suite 3.3 tiene una nueva consola basada en web diseñada específicamente para técnicos con experiencia mínima. Complementa la consola completa y es fácil de usar con asistentes integrados para tareas comunes como la migración de Windows 10 y la implementación de imágenes. Tiene una interfaz de usuario moderna y funciona con cualquier navegador. Con la consola web, los trabajos ejecutados con frecuencia se pueden delegar a los técnicos de primera línea para optimizar los recursos y reducir los costos. La consola web también es altamente escalable, por lo que una gran cantidad de técnicos pueden usarla simultáneamente.

La nueva consola web incluye asistentes guiados para acortar la curva de aprendizaje de imágenes para administradores sin experiencia. Ghost Solution Suite 3.3 tiene asistentes para crear trabajos para una actualización in situ a Windows 10, y para implementar y configurar imágenes. Se planean asistentes adicionales para los próximos lanzamientos.

El uso de los asistentes toma lo que pueden ser procesos complejos y los transforma en simples flujos paso a paso que harán que incluso el administrador más nuevo se sienta como un profesional.

Una parte importante del tiempo requerido para las máquinas de imágenes es iniciar la automatización. Cuanto más demore una máquina en ingresar a la automatización, más tardará el proceso general de generación de imágenes. Ghost Solution Suite 3.3 ahora es compatible con iPXE, que ofrece una mejora espectacular de la velocidad en el proceso de automatización previo al arranque. iPXE acorta el tiempo de inactividad general durante las actividades de implementación y, por lo tanto, hace que el usuario final recupere rápidamente la productividad.

Las características de esta herramienta son:

- Es uno de los mejores herramientas para captura e implementación de imágenes de disco.
- Permite la instalación de sistema operativo mediante un *script*.
- Permite la migración de configuraciones de usuarios y sus aplicaciones.
- Dispone de un sistema inteligente de distribución de controladores de dispositivo.
- Permite la ejecución de tareas remotamente y en secuencia.
- Tiene soporte para los sistema operativos: Windows, Mac y Linux.
- Permite un gran nivel de automatización del entorno.

2.1.2. Acronis Snap Deploy 5

La información mostrada se ha obtenido de la hoja de datos del producto [2].

Acronis Snap Deploy es la solución idónea para el despliegue rápido de inicio sin sistema operativo (bare-metal) y para un gran numero de estaciones de trabajo y servidores, así como el despliegue sobre el mismo hardware en educación, formaciones y entornos de laboratorios. Acronis Snap Deploy le permite aprovisionar cientos de estaciones de trabajo y servidores a la misma velocidad que si fuera uno solo. Basado en Acronis AnyData Engine y la galardonada tecnología de imagen de discos, le permite crear una imagen de disco de cualquier configuración estándar que elija, incluyendo el sistema operativo, la configuración, los ficheros y todas las aplicaciones, y, simultáneamente, desplegarla en múltiples equipos en un solo paso. Basada en más de 100 patentes, Acronis AnyData Engine es el core de la suite que se compone de todos los productos de nueva generación de Acronis, permitiendo capturar, almacenar, recuperar, controlar y acceder a los datos. Incluye Acronis® Universal Deploy®, que le permite desplegar la misma imagen a otros PCs y servidores incluso si son de distinto fabricante. El producto gestiona también configuraciones específicas de equipos, incluyendo el nombre, las direcciones IP, la pertenencia a dominio y otras configuraciones. Además al finalizar el despliegue, Acronis Snap Deploy le permitirá modificar de forma sencilla la configuración de Windows.

Las características de esta herramienta son:

- Despliegue de gran velocidad. Desde el punto central de administración, podrá de forma rápida desplegar una imagen a múltiples PCs o servidores vía unicast o multicast.
- Sencillo de utilizar. El interfaz simplificado de usuario ahorra tiempo y reduce la posibilidad de errores.
- Múltiples sistemas operativos. Proporciona flexibilidad permitiendo desplegar Windows Server®, equipos de sobremesa, tabletas basadas en Windows tablets, y Linux.
- Opciones de despliegue flexibles. Puede escoger entre 4 opciones de despliegue, incluyendo, manual, automática, iniciada por el usuario y planificada. Podrá desplegar una imagen en modo independiente incluso si el equipo no está en la red.
- Despliegue en cualquier Hardware. Se puede desplegar la misma imagen en un hardware distinto, incluso en múltiples equipos basados en Intel al mismo tiempo.
- Configuraciones específicas para equipos. Puede asignar configuraciones individuales para cada PC o servidor para garantizar que reciben la configuración correcta. Podrá monitorizar el progreso y el estado mediante la lista de “nuevos equipos” y pre-configurar los nuevos equipos mediante su dirección MAC.
- Wake-On-LAN. Puede arrancar de forma automática sus PCs para el despliegue con Wake-on-LAN, asegurando que se lleva a cabo acorde a la programación.
- Licenciamiento flexible. Las opciones de licenciamiento incluyen por despliegue, por equipo, y por site para instalaciones más grandes.

2.1.3. SmartDeploy

La información mostrada se ha obtenido de la web del desarrollador del producto [3].

Con SmartDeploy, sus administradores pueden crear una imagen maestra del sistema operativo e implementarla en escritorios y servidores desde una ubicación central.

Las características de esta herramienta son:

- Implemente imágenes, aplicaciones, controladores y actualizaciones en máquinas con licencia una vez o un millón de veces.
- Las capas de imagen base independientes del hardware se pueden implementar en cualquier marca y modelo de computadora.
- Cree ilimitados archivos de respuesta específicos del departamento con un asistente intuitivo para implementaciones rápidas y consistentes fácilmente.
- Instale SmartDeploy en el equipo que ya tiene. No es necesario comprar, configurar y mantener hardware costoso o complicado.
- SmartDeploy es compatible con Windows 7 a través de Windows 10 y Server 2012 y superiores.
- Implementaciones que van desde la nube hasta completamente fuera de línea.
- Guía sencilla para ayudarte en cada paso.

- Los asistentes simples paso a paso proporcionan una guía infalible para completar sus proyectos.

Los servicios que incluye este software son:

- Administre computadoras remotas usando su almacenamiento en la nube Box, Dropbox, Google Drive o OneDrive. No se necesita conexión VPN.
- Migración de datos de usuario con un clic. Conserve los datos del usuario al volver a crear su dispositivo existente o al configurar uno nuevo.
- Mantenga automáticamente los controladores de dispositivos actualizados en sus puntos finales con monitorización en tiempo real y actualización de controladores con un solo clic.
- Disfrute de acceso ilimitado a nuestra biblioteca de paquetes de aplicaciones y cree sus propios paquetes de aplicaciones personalizados para la implementación a sus usuarios.
- multicast. Implemente en 25 o más dispositivos a la vez sin saturar su red.
- Ver información sobre las máquinas de la red.
- Implementación remota de toque cero. Envíe imágenes, aplicaciones, controladores y actualizaciones a través de su red existente o almacenamiento en la nube desde su Consola de administración centralizada.
- Nuestros técnicos crean y mantienen paquetes de controladores de dispositivos específicos del modelo.
- Implemente paquetes de aplicaciones desde nuestra biblioteca o cree los suyos para implementarlos en su red existente o en la nube.
- Elija de nuestra biblioteca o cree sus propios scripts para implementar individualmente o como parte de su imagen.
- Mueva las licencias de SmartDeploy a una computadora nueva cuando quite o reemplace una computadora de su organización.
- Las nuevas versiones incluyen nuevas características, funcionalidad avanzada y soporte para las últimas versiones de Win10.

2.1.4. CloneDeploy

La información mostrada se ha obtenido de la web del desarrollador del producto [4].

CloneDeploy es una solución gratuita de código abierto para la clonación y creación de imágenes de una computadora. Es una alternativa para otras soluciones, como Symantec Ghost Solution Suite, Acronis Snap Deploy, Servicios de implementación de Windows, etc.. Admite imágenes de Windows XP a través de Windows 10 y Linux, utilizando tecnologías unicast y Multicast.

Las características de esta herramienta son:

- Múltiples entornos de imágenes para una mayor compatibilidad.

- Compatibilidad con unidades nativas 4K, compatibilidad con unidades de formato avanzado y compatibilidad con unidades 512b.
- Implemente imágenes EFI en equipos heredados y viceversa (actualmente solo imágenes de Windows).
- Soporta MBR y GPT.
- Soporta Legacy BIOS and EFI.
- Soporte completo de LVM.
- Soporte completo de partición extendida / lógica.
- Soporta formatos NTFS, FAT, ext2, ext3, ext4, xfs.
- Clonación a unidades más pequeñas.
- Cambio automático de nombre de computadora.
- Arranque del cliente desde USB / ISO / PXE.
- Interfaz web sencilla.
- Imágenes completamente automatizadas.
- Secuencias de comandos personalizadas.
- Soporta multicast.
- Se puede instalar en Windows o Linux.

2.1.5. Kace SDA

La información mostrada se ha obtenido de la web del desarrollador del producto [5].

Realiza rápida y fácilmente el aprovisionamiento inicial y la administración continua de imágenes del sistema y las actualizaciones de controladores en diferentes plataformas de hardware, para reducir los costos operativos y garantizar la productividad del usuario final. KACE Systems Deployment Appliance automatiza la implementación de sistemas a gran escala y simplifica las migraciones de múltiples sistemas operativos, por lo que tendrá más tiempo para enfocarse en otros proyectos de alta prioridad. Con KACE SDA , puede hacer lo siguiente:

- Ahorrar tiempo con los procesos automatizados de creación de imágenes e implementación de copias maestras principales.
- Realizar rápidamente implementaciones masivas de sistemas en sitios remotos desde cualquier ubicación.

Las características de esta herramienta son:

- Implementación y creación de imágenes del sistema. Ahorre tiempo y simplifique la implementación y la creación de imágenes de disco en Windows® y Mac®.
- Implementaciones automatizadas.
- Administración de sitio remoto.

- Multidifusión.
- Administración de estado de usuario de Windows. Simplifique las migraciones al implementar de manera automática ajustes y archivos específicos del usuario junto con el sistema operativo y las aplicaciones, y prácticamente elimine el riesgo de perder información crítica.
- Administración. Obtenga más tiempo en el día y reduzca la complejidad al abordar todas las necesidades de recuperación, implementación y aprovisionamiento de su sistema mediante una única solución.

2.1.6. Fog Project

La información mostrada se ha obtenido de la web del desarrollador del producto [6].

Es una solución gratuita. Fog Project es una aplicación que despliega de forma remota imágenes de discos (sistemas operativos) de ordenadores, haciendo ideal su uso en ambientes o situaciones empresariales o educativas donde se requiera facilitar el clonado de equipos en breve tiempo. Cuenta con una interfaz web amigable que permite administrar distintos equipos con distintas imágenes y desplegarlas por red (en forma unicast como multicast). Sus características son:

- Puede capturar, implementar y administrar Windows, Mac OSX y varias distribuciones de Linux.
- Las computadoras se pueden administrar de forma segura con FOG Project de forma remota, desde cualquier parte del mundo (para ello se requiere de un servidor FOG público).
- Imágenes independientes del hardware.
- Administración remota de computadoras.
- Instalaciones multisitio.
- Administración mediante una interfaz web.
- Automatización del trabajo.

2.1.7. OpenGnsys

La información mostrada se ha obtenido de [10].

Es una aplicación formada por una serie de herramientas libres y abiertas que constituyen un sistema completo, versátil e intuitivo, para la gestión y clonación de equipos. Este sistema permite la distribución, instalación y despliegue de distintos sistemas operativos. Posee una arquitectura lo suficientemente flexible para adaptarse a las necesidades de los diferentes modelos de redes de ordenadores, tanto de empresas como de instituciones, pudiendo ser utilizado en distintos tipos de escenarios:

- Gestión centralizada de Unidades TIC de Apoyo a Docencia e Investigación.
- Mantenimiento del parque de ordenadores personales de los miembros de una Institución.

- Despliegue y mantenimiento de los servidores de un Centro de Datos en particular.
- Administración de repositorios de clonación de equipos para los Servicios de Soporte (helpdesk) de una Institución.

Las principales características de OpenGnsys son:

- Trabaja a nivel de sistema de ficheros.
- Permite restaurar en particiones de menor tamaño que la inicial.
- Permite la configuración del sistema operativo en el cliente destino.
- Dispone de una consola centralizada y un menú de inicio en los clientes.
- Sencillez y flexibilidad.
- Empleo de una interfaz web para su administración.
- Detección de inventario de hardware y software.
- La distribución de imágenes y ficheros puede realizarse mediante unicast, multicast y P2P e incluso off-line.
- Despliegue en sistemas Windows, Linux y Mac OS.

2.2 Comparativa de las herramientas anteriores

A continuación se muestra la tabla 2.1 con una comparativa de las herramientas comentadas en la sección 2.1.

	Ghost	Acronis	SmartDeploy	CloneDeploy	Kace	FOG	OpeGnsys
Consola centralizada	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Despliega Windows	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Programación despliegue	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Wake-OnLan	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Pago por licencias	Sí	Sí	Sí	No	Sí	No	No

Tabla 2.1: Comparativas de las herramientas de la sección 2.1

Inicialmente todas las herramientas que se han mostrado cumplen los requisitos necesarios. Probablemente teniendo en cuenta sus características, las herramientas mejores sean:

- Symantec Ghost Solution Suite.
- Acronis Snap Deploy 5.

El inconveniente que presentan estas herramientas es su alto coste en el licenciamiento para su utilización. Teniendo en cuenta que los presupuestos que gestiona un departamento son reducidos y, teniendo alternativas que cumplen los requisitos y que además son gratuitas, se descartan todas las herramientas que tengan coste de licenciamiento.

Realizando un estudio más a fondo de las características de las herramientas gratuitas se decide utilizar **OpenGnsys** por una serie de motivos que comento a continuación:

- Actualmente en el AIM se trabaja con equipos Windows. Se dispone de uno o dos equipos Linux y algún equipo con el sistema operativo Mac OS. Es interesante disponer una herramienta que soporte las tres plataformas.
- El coste económico de las licencias de utilización (es gratuito). Existen herramientas muy completas para realizar el clonado de los equipos de una organización pero tienen un alto coste de licenciamiento para poder utilizarlas.
- Clonación de sistemas operativos Windows sin necesidad de recurrir a la utilidad Sysprep de Microsoft, lo que simplifica mucho la creación de una imagen base de Windows.
- Distribución de las imágenes mediante protocolos unicast, multicast o bittorrent (P2P). Estos dos últimos son especialmente interesantes cuando tenemos que hacer la distribución a muchos equipos a la vez.
- Su uso es sencillo.
- Es modular y escalable.
- Es una herramienta flexible y robusta.
- Permite la administración delegada.
- Post configuración de sistemas Windows y Linux mediante *scripts* en Bash Shell.
- Es un proyecto vivo y en constante evolución.
- Gran equipo detrás del desarrollo, buena documentación y foro activo frente a dudas y problemas que van surgiendo. Si además de esto, se tiene en cuenta que un compañero de esta universidad está muy involucrado en el proyecto participando activamente, es un valor añadido. La comunicación se hace mucho más activa y directa frente a problemas y soluciones.
- Apuesta por parte de esta universidad por esta herramienta. Se ha aumentado considerablemente el número de centros que emplean activamente esta herramienta desde hace tiempo.
- Proyecto RemotePC. Es un motivo más para la elección de esta herramienta.
- Uso en diferentes redes de área local virtual (VLAN).

Sin duda es una opción excelente para atajar la necesidad del despliegue de imágenes del sistema operativo en las aulas del AIM, ya no solo enfocado a las aulas informáticas sino que también se puede ampliar al resto de equipos informáticos del área (servidores, equipos profesorado, equipos de laboratorios, equipos de becarios y doctorandos, etc.).

Una vez elegida la herramienta que se va a utilizar para poder desplegar imágenes del sistema operativo, se expone el diseño de la solución que se propone. Para ello se ha de realizar el proceso inicial de preparación de un equipo base sobre el cual se va crear una imagen que será desplegada al resto de equipos de las aulas informáticas. Los pasos necesarios para esa creación inicial son:

- Crear en OpenGnsys una unidad organizativa sobre la que se va a trabajar.
- Añadir el equipo base a dicha unidad organizativa.

- Configurar a través de la BIOS del equipo que se va a utilizar como base para que arranque a través de la red (PXE).
- Particionar el disco del sistema de equipo base con OpenGnsys.

En estos momentos se va a realizar la preparación del equipo, para lo cual se realizan una serie de pasos:

- Realizar la instalación del sistema operativo (en nuestro caso Windows 10 64 bits).
- Instalar los drivers necesarios del equipo.
- Realizar las actualizaciones pertinentes del sistema operativo (Windows Update).
- Realizar un ajuste del sistema (modificación de la memoria virtual, desactivar servicio Windows Search, chequeo del disco duro, eliminación de aplicaciones de la tienda de Microsoft innecesarias, ajuste de opciones de rendimiento del equipo, etc.).

Se genera una imagen inicial de este equipo (sería la imagen base del sistema) a través de OpenGnsys. Teniendo en cuenta que podemos disponer de equipos con una configuración hardware diferente, se ha de reajustar dicha imagen para añadir los drivers de equipos adicionales. Es por ello que se ha de restaurar la imagen base que se ha generado en un principio en cada equipo con un hardware diferente. Posteriormente se añadirán los nuevos drivers necesarios para ese equipo. Finalmente, se ha de generar otra imagen que sobrescriba la imagen base inicial que se ha creado. Este proceso se ha de repetir tantas veces como equipos con hardware diferente se disponga. Concluido este proceso ya se tendrá una imagen base con todos los drivers que se necesiten en las diferentes aulas.

Ya con la imagen base perfectamente actualizada, se procede a restaurarla en un equipo de las aulas. El siguiente paso es instalar todo el software que vaya a ser necesario para su utilización en las prácticas docentes. Una vez finalizado el proceso de instalación del software se ha de realizar una pequeña revisión y ajuste para verificar que todo es correcto. Antes de proceder a la generación de la imagen de este sistema perfectamente configurado, es necesario la realización de eliminación de archivos temporales en el sistema. Alcanzado este punto, ya se está en disposición de genera una imagen completa con el software necesario. Será esta segunda imagen la que se va a desplegar en todos los equipos de las aulas informáticas del AIM.

Se ha de indicar que sobre la imagen base que se generó inicialmente se va actualizando de vez en cuando para aplicar las actualizaciones de Microsoft y algún ajuste adicional. Será sobre esta base inicial sobre la que se realizará el proceso de instalación de todo el software necesario para la preparación de la imagen que se replicará a todos los equipos necesarios cada vez que sea necesario un proceso de clonación del sistema operativo (como se ha comentado en la actualidad se va a realizar al menos una o dos veces al año).

Teniendo la imagen del sistema preparada lo que se ha de realizar es el despliegue de dicha imagen a los equipos de las aulas. Para ello se realizarán una serie de pasos:

- Ya tenemos implementada en OpenGnsys nuestra estructura lógica del AIM.
- Se realiza un procedimiento de preparación de los discos donde se va a realizar el clonado (se empleará un procedimiento para ello).

- Realizar un procedimiento de despliegue de la imagen sobre los equipos que queramos. Se ha de elegir el método de despliegue que se desee (unicast, multicast, empleo de la cache o P2P).
- Se realiza el proceso de configuración posterior al despliegue de la imagen (realizando tareas como cambio de nombre en el equipo, añadir el equipo en el dominio, antivirus, etc.).

Acabado todo el procedimiento expuesto se tendrá en todos los equipos de las aulas informáticas la misma configuración del sistema operativo que es lo que se pretende.

CAPÍTULO 3

Tecnología de clonado

En un equipo informático puede pasar de todo: un virus, un troyano, etc. que pueden alterar el buen funcionamiento del sistema. La instalación o actualización de un software o un driver puede dejar el sistema inestable, o en el peor de los casos, dejarlo fuera de servicio. Una descarga eléctrica puede destruir los componentes del mismo (placa base, memoria, etc.).

Si pensamos en el tiempo que se tendría que invertir en la reinstalación del sistema operativo, la instalación de los periféricos y todo el software necesario junto con la configuración personalizada del sistema y las aplicaciones, protección del sistema, parches, etc. sería considerable. Pensemos además que esto sería para un único equipo. ¿Qué pasaría si nos ocurriera con un montón de ellos?. Tendríamos claro que no se podría realizar esta operación equipo a equipo y se hace necesario un sistema más eficiente y menos costoso. Surge la necesidad del uso de las tecnologías de clonado. Pero, ¿qué se entiende por clonación de un equipo informático?. Consiste en generar un duplicado exacto de la información que tenemos en un disco duro de un ordenador (a otro disco duro o a un archivo imagen) con el fin de restaurar dicho contenido en uno o más ordenadores. Las ventajas del clonado de discos son claras: copia de seguridad, instalación de una imagen en múltiples equipos, mantenimiento, sustitución de un disco duro por otro de mayor tamaño, etc..

En castellano según la RAE clonar es [8] Producir clones y clon solo hace referencia a copias biológicas. Sin embargo, en inglés se tiene una acepción en el verbo *clone* que es producir una copia de, y esa copia puede ser o no biológica. Por tanto, en inglés se puede clonar un coche pero en castellano no. A pesar que en el lenguaje castellano no es posible clonar equipos informáticos y está mal dicho, conscientemente se utiliza esta expresión en el ámbito tecnológico y se entiende el significado.

Según [9], las herramientas informáticas empleadas para la clonación se clasifican dependiendo de su capacidad para acceder al sistema operativo en :

- Clonación basada en archivos.
- Clonación basada en el contenido del sistema de archivos.
- Clonación basada en el contenido de la partición.
- Clonación basada en sectores: disco o partición.

La **clonación basada en archivos** permite mejorar los procesos de clonación basados en el contenido de los sistemas de archivos, debido a que trabaja con las diferencias de los archivos. Se consigue disminuir los tiempos de los procesos de creación y restauración de imágenes y el espacio de almacenamiento necesario para guardar las imágenes.

La **clonación basada en el contenido del sistema de archivos** ofrece la ventaja de copiar solo los datos de usuario y considera el tamaño de los datos como el sistema de archivos a clonar (independientemente de la partición que la incluye). El tamaño de la partición destino no tiene que tener el tamaño de la partición de origen; debe ser al menos igual a los datos de usuario.

La **clonación basada en el contenido de la partición** tiene la ventaja de trabajar únicamente con datos útiles (no se tiene en cuenta el espacio no utilizado). Este hecho, hace que aumente la rapidez en el proceso de clonado. Como inconveniente de este sistema de clonación, el tamaño mínimo del dispositivo de destino ha de tener al menos el mismo tamaño que la partición de origen.

La **clonación basada en sectores** duplica cada uno de los sectores del dispositivo de almacenamiento. Como principales características de este tipo de clonación, se tiene:

- Requiere mayor tiempo en el proceso de clonado.
- Necesita un tamaño mínimo en el dispositivo de destino (al menos el mismo número de sectores que en el origen).

Es probable que nos hagamos la siguiente pregunta: ¿cómo funcionan realmente las tecnologías de clonado?. Lo primero que hay que hacer es preparar un equipo base o modelo que queremos clonar para replicar en uno a varios equipos. Para ello se debe realizar una serie de pasos:

- Particionar el disco del sistema adecuadamente.
- Instalación del sistema operativo.
- Actualización de los drivers de los componentes físicos del equipo. Actualización del sistema operativo con los parches de seguridad necesarios que mejoran su estabilidad.
- Instalación de las aplicaciones de usuario necesarias.
- Configuración de las aplicaciones y del sistema operativo.
- Verificar el correcto funcionamiento del sistema.

Una vez se haya comprobado que todo es correcto y que el sistema está estable y funciona óptimamente, se necesita hacer una imagen del sistema. Una vez tengamos la imagen del sistema modelo, será necesario restaurarlo a múltiples ordenadores (esta acción se conoce habitualmente como despliegue -Deploy-).

Se podría pensar que es algo trivial y sencillo pero no lo es. ¿Por qué?. Pues porque no siempre se dispone de equipos idénticos con la misma configuración hardware. Alguien podría imaginar que cuando tenemos un equipo que no funciona (no arranca por un problema hardware) sería tan sencillo como extraer el disco duro del sistema y ponerlo

en otro ordenador. Muy probablemente no funcione por una serie de motivos entre los que se encuentran los siguientes:

- El hardware del ordenador no tiene por qué ser igual.
- Los controladores de los dispositivos que hay instalados en un ordenador no tienen por qué funcionar en otro. Ha de ser el sistema operativo el que necesita explorar el equipo para determinar los controladores que son necesarios para cada dispositivo del sistema.
- Dentro de una red local, no podemos tener dos equipos con el mismo nombre porque provocarían conflictos a la hora de identificación del equipo.

Se hace necesario que tras la restauración del sistema en uno o varios equipos se realice un proceso automático o semiautomático de postconfiguración que nos permita cambiar el nombre del equipo, cambiar el SID de la máquina (en sistemas operativos Microsoft se emplea el uso de *Sysprep*), la posibilidad de meter en en el dominio al equipo restaurado, etc.) y esto las tecnologías de clonado lo tienen también.

Teniendo claro en qué consisten las tecnologías de clonado necesarias para la gestión de equipos informáticos en organizaciones con un tamaño importante de equipos, existen un montón de herramientas que desempeñan estas funcionalidades.

CAPÍTULO 4

OpenGnsys

El Proyecto OpenGnsys [10] (nace en mayo de 2009) reúne el esfuerzo conjunto de varias Universidades Públicas Españolas con el fin de disponer de una serie de herramientas libres y abiertas que constituyan un sistema completo, versátil e intuitivo, para la gestión y clonación de equipos. Este sistema permite la distribución, instalación y despliegue de distintos sistemas operativos.

Posee una arquitectura flexible que permite adaptarse a las necesidades de los diferentes modelos de redes de ordenadores, tanto de empresas como de instituciones, pudiéndose utilizar en distintos tipos de escenarios:

- Gestión centralizada de Unidades TIC de Apoyo a Docencia e Investigación.
- Mantenimiento del parque de ordenadores personales de los miembros de una Institución.
- Despliegue y mantenimiento de los servidores de un Centro de Datos en particular.
- Administración de repositorios de clonación de equipos para los Servicios de Soporte (*helpdesk*) de una Institución.

El Proyecto OpenGnsys es un sistema modular (tal como se observa en la figura 4.1) integrado por un conjunto de componentes interrelacionados entre sí que se pueden adaptar a una gran cantidad de escenarios de trabajo posibles. Estos componentes son:

- **Servidor OpenGnsys.**
Lo conforma un conjunto de servicios básicos del sistema de clonación necesarios para el arranque y configuración inicial de los clientes.
- **Administrador web OpenGnsys.**
Se trata de una consola web que posee una interfaz muy sencilla permitiendo el control de todos los componentes del sistema. Se puede realizar una gestión centralizada del sistema en su conjunto posibilitando la delegación de la administración de una parte del sistema.
- **Gestor del repositorio OpenGnsys.**
Contenedor donde se almacenan las imágenes para cada una de las unidades administrativas que se han definido en el módulo de administración de OpenGnsys. En función del modelo de organización definido para un sistema, pueden existir uno o varios repositorios.

- **Ciente OpenGnsys -ogLive-.**

Es un Agente que ejecuta órdenes en el cliente. Posee las siguientes funcionalidades:

- Posee una interfaz gráfica desde donde se elige de entre las opciones proporcionadas por el administrador.
- En modo administrador, permite la ejecución interactiva de funciones del motor de clonación.
- Permite la ejecución de cuantas tareas se envíen desde el módulo de administración de OpenGnsys.

- **Motor de clonación OpenGnsys.**

Contiene una serie de librerías de funciones utilizadas para la instalación y el arranque de los sistemas operativos en el cliente.

- **Instalador OpenGnsys.**

Son una serie de programas para se utilizan para la instalación, actualización y desinstalación del sistema.

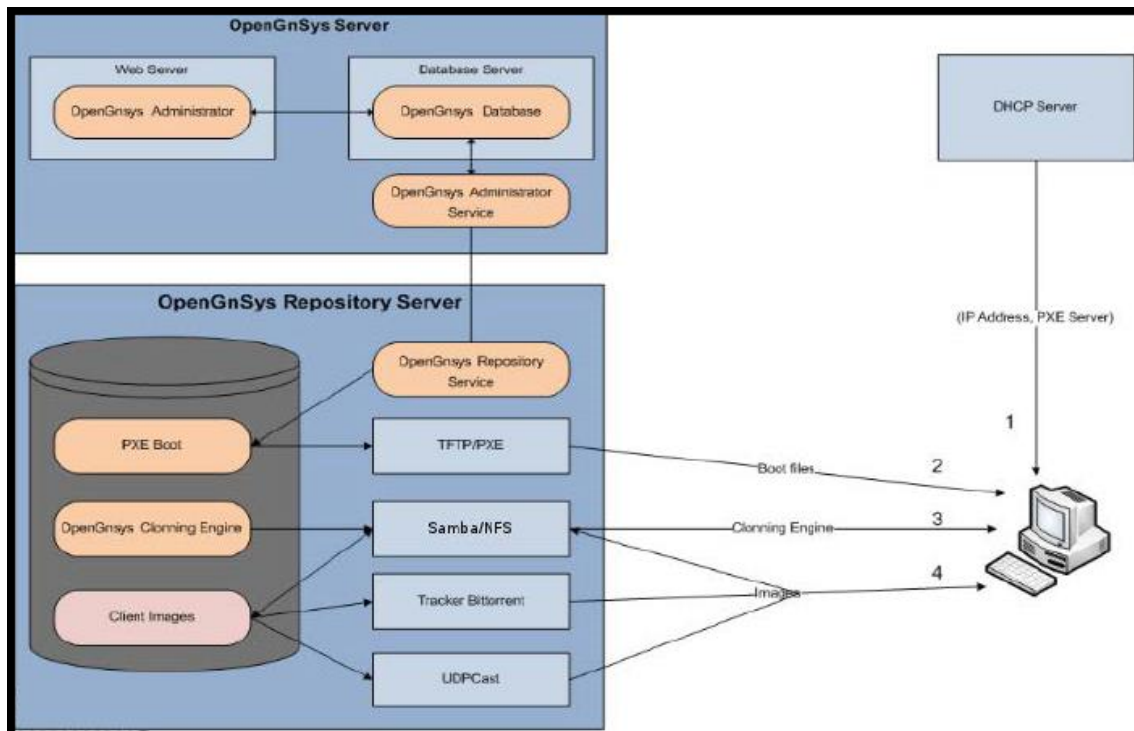


Figura 4.1: Arquitectura de componentes de OpenGnsys

Los módulos comentados con anterioridad están separados en distintas capas de servicios, que son:

- Capa inferior, encargada del acceso directo a los dispositivos del cliente y de las funciones que provee el motor de clonación.
- Capa intermedia, constituida por un conjunto de herramientas que permite realizar tareas complejas y personalizar el entorno.
- Capa de administración, formada por una interfaz web y una base de datos de gestión.

Entrando un poco en materia, lo primero que se necesita es disponer de un equipo modelo o base donde se ha instalado uno o más sistemas operativos con los programas y datos que necesitemos. Una vez verificado (en este equipo modelo) el correcto funcionamiento del sistema y de las aplicaciones instaladas y configuradas, se procede a la creación de imágenes de sus sistemas de ficheros que más tarde se van a emplear para replicarlas a todos los ordenadores de nuestra organización o a un grupo de ellos.

La distribución de las imágenes y ficheros se puede realizar de una forma flexible mediante una serie de protocolos de comunicaciones que se ha implementado: unicast [11], multicast [12] y P2P [13]. Además es posible disponer de dicha información de manera fuera de línea (*off-line* que es el anglicismo equivalente) sin necesidad de comunicarse con el servidor (o bien se accede directamente a la caché de datos local de cada cliente o se accede a un dispositivo externo de almacenamiento).

La restauración del sistema operativo se realiza a nivel de sistema de archivos y supone una copia exacta del modelo base. El tamaño de la partición destino puede ser menor que el tamaño de la partición original (tiene que tener como mínimo el tamaño ocupado por los datos más 1 GB adicional). Tras la restauración se pueden realizar ciertas tareas de configuración en el equipo destino: cambio de nombre del equipo, detección del inventario de hardware [14] y software [15] instalado, copia o borrado de ficheros y directorios, modificación del registro Windows, inclusión en un dominio, etc.. Estas tareas específicas se pueden realizar sin necesidad de arrancar el sistema operativo mediante el acceso a la información de los discos.

Habiendo realizado una pequeña introducción a la herramienta OpenGnsys (qué es, sus características y arquitectura), lo primero que se ha de realizar es su instalación.

4.1 Preparación de la máquina servidora

Los requisitos hardware del equipo donde se ha de instalar el servidor OpenGnsys no son elevados: se necesita poco disco duro [16] y memoria. Incluso se puede optar por realizar la instalación en un equipo virtualizado. Para la instalación del servidor que va a albergar OpenGnsys se ha empleado un equipo con una placa base *Intel DH61BE* con un procesador *Intel(R) Core(TM) i5-2400 a 3.10 GHz* y 8 GB de memoria DDR3. Posee dos discos duros: uno de 80 GB que contendrá el sistema operativo y aplicaciones y otro de 2 TB que contendrá las imágenes de los sistemas operativos que queramos clonar.

4.1.1. Instalación del sistema operativo

Se ha consultado la lista de compatibilidad para la instalación del servidor OpenGnsys [17]. Teniendo esto presente, se ha procedido a descargar la imagen ISO [18] de **Ubuntu Server 16.04 Xenial de 64 bits** desde el repositorio de CICA [19].

Se ha procedido a generar una memoria USB autoarrancable con la imagen ISO descargada mediante el software Rufus [20]. Se debe entrar en la BIOS del equipo donde se va a instalar el servidor OpenGnsys para cambiar las opciones de arranque de modo que se haga desde un soporte USB [21].

Se procede con la instalación del sistema operativo Ubuntu Server tal como se puede observar en las tablas 4.1, 4.2 y 4.3 donde se ha ido siguiendo las instrucciones del asistente de instalación que comienza con la selección del idioma, el país donde nos encontramos, la configuración del teclado, la configuración de red que por defecto se proporciona de manera automática por el servidor DHCP de la universidad (previamente se ha dado de alta el equipo en el sistema de red propio de la universidad), el nombre de la máquina, el usuario y la contraseña para entrar en el sistema, el particionado del disco (inicialmente se dispone de un disco SATA de 80 GB de capacidad) donde se ha utilizado el modo guiado empleando toda la capacidad del disco, no se va a utilizar ningún servidor proxy y no se van a instalar actualizaciones automáticas del sistema. Como software adicional se marca la opción de servidor OpenSSH y finalmente se instala el cargador de arranque en el gestor de arranque. Una vez reiniciado el equipo se observa en la última viñeta que nos pide un usuario para entrar en el sistema operativo recién instalado.

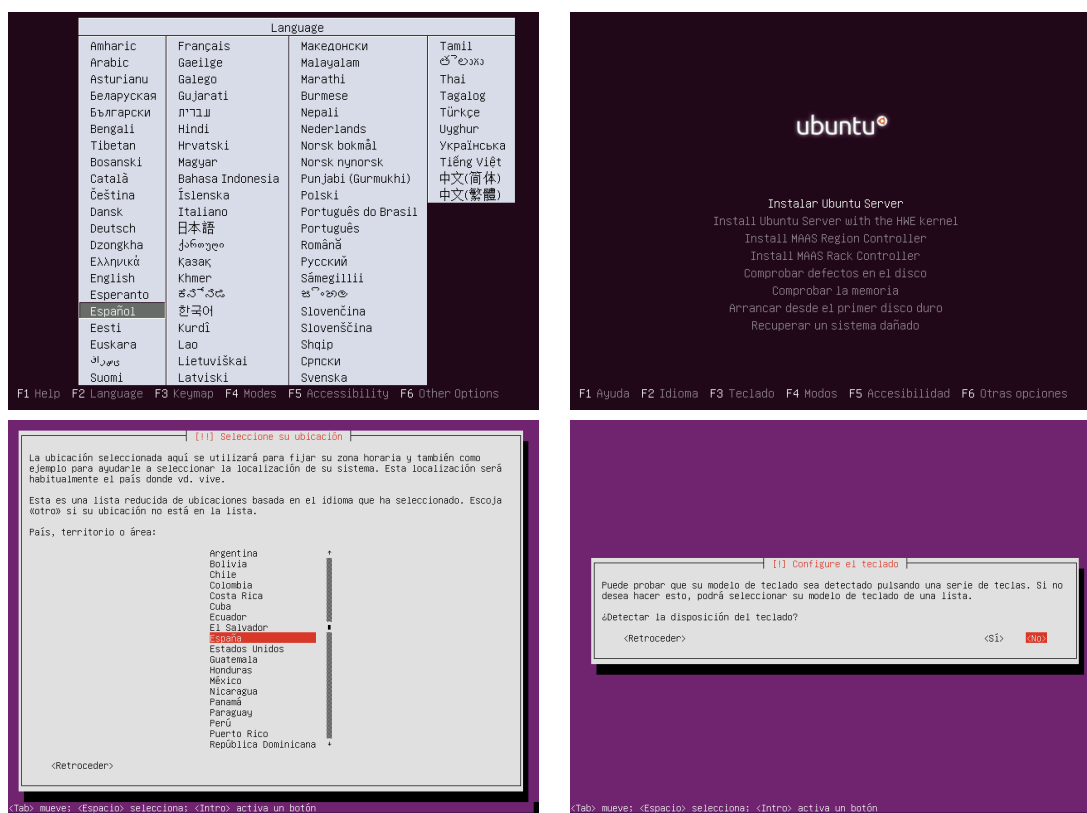


Tabla 4.1: Instalación Ubuntu Server (Viñetas 1-4) ¹

¹Las viñetas se han puesto de arriba a abajo y de izquierda a derecha

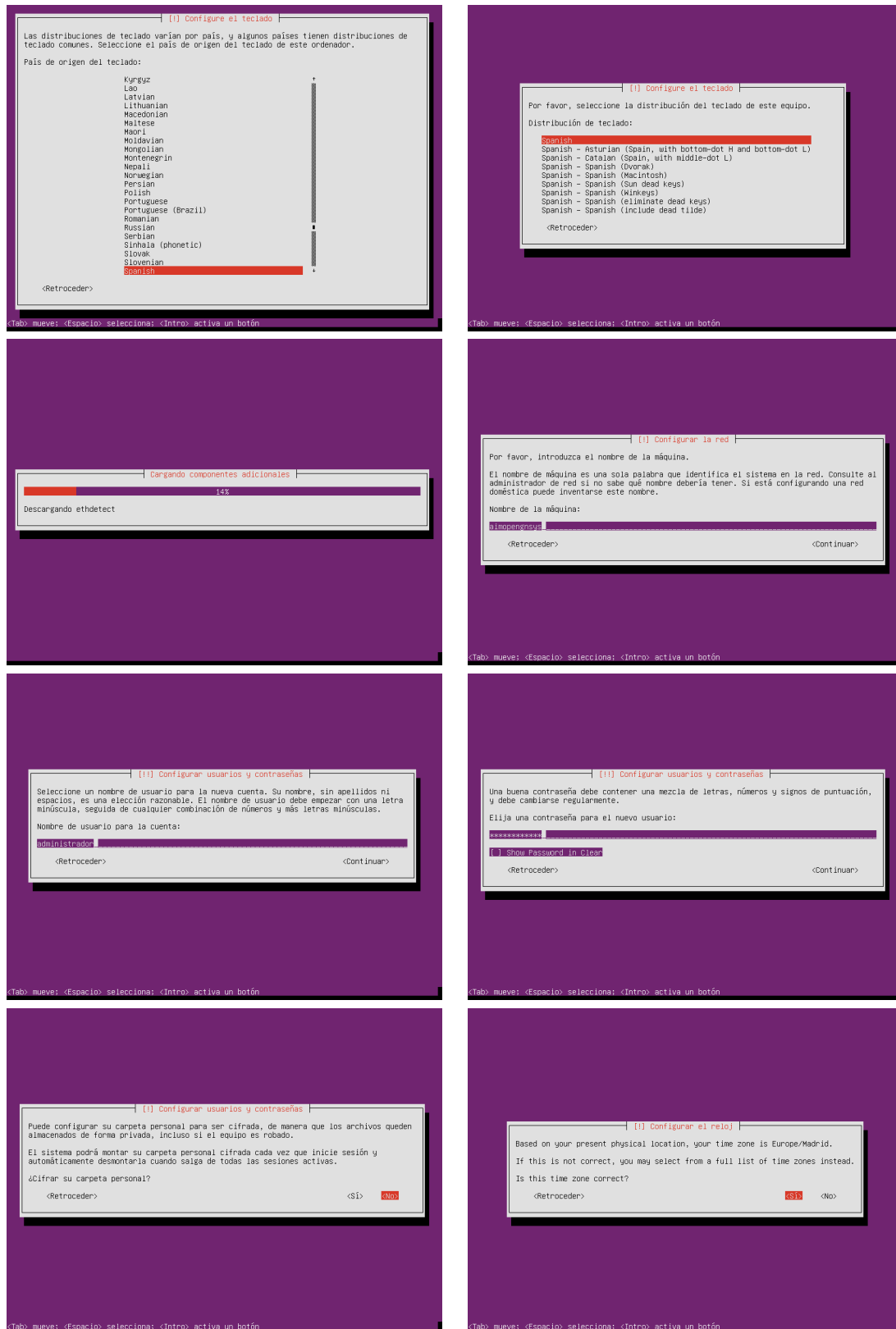


Tabla 4.2: Instalación Ubuntu Server (Viñetas 5-12)²

²Las viñetas se han puesto de arriba a abajo y de izquierda a derecha

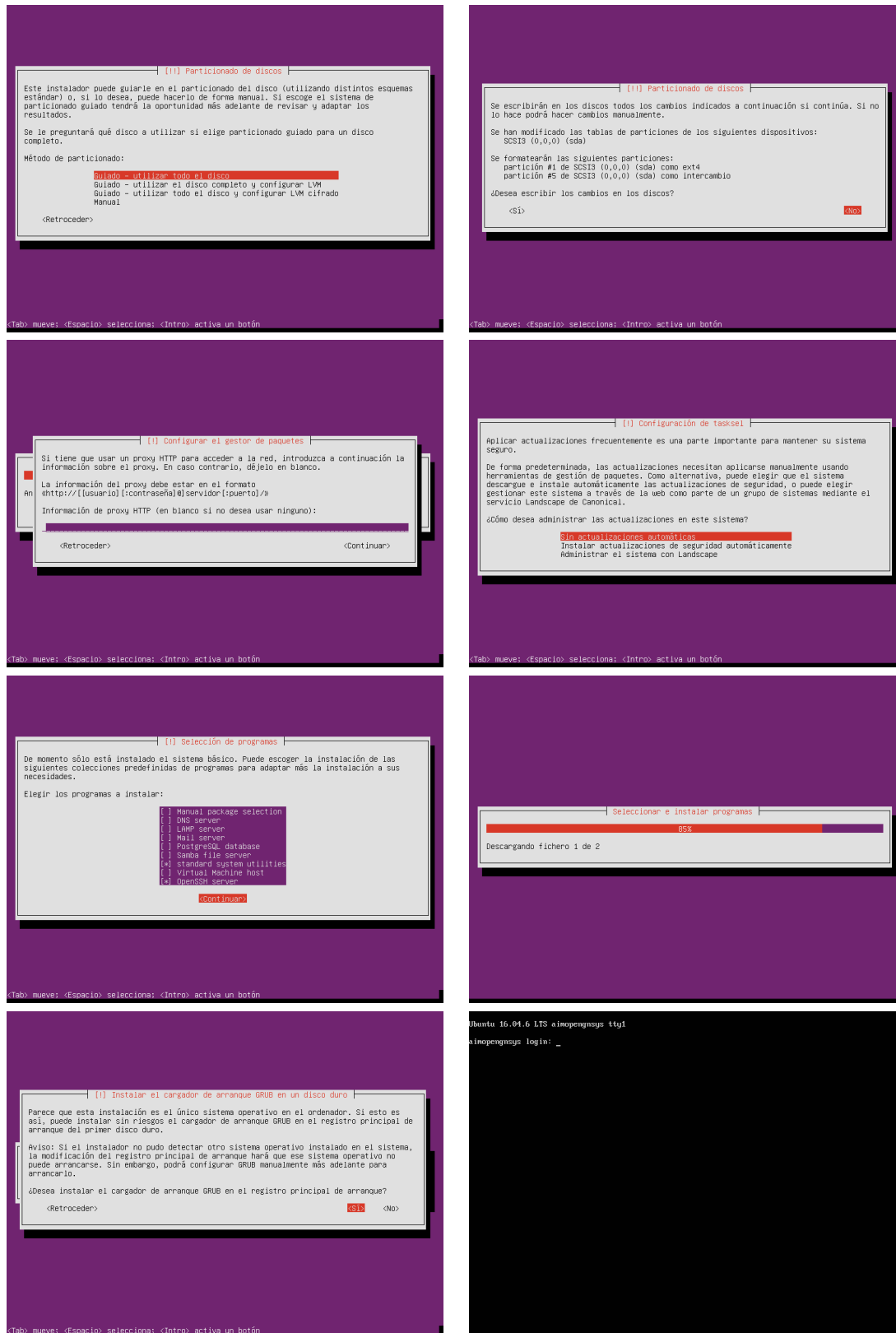


Tabla 4.3: Instalación Ubuntu Server (Viñetas 13-20) ³

Una vez se ha instalado el sistema operativo se puede conectar de una manera sencilla y segura desde nuestro equipo Windows a este equipo desde una consola SSH [22]. Para ello se ha empleado la aplicación MobaXterm Home Edition [23] que se ha descar-

³Las viñetas se han puesto de arriba a abajo y de izquierda a derecha

gado desde la web del desarrollador. Mediante esta aplicación se va a trabajar con mayor comodidad.

4.1.2. Creación de una conexión SSH al servidor OpenGnsys

Desde la aplicación MobaXterm vamos al menú **Sessions =>New session**. De esta manera se nos abre una nueva ventana donde nos aparece la configuración de la sesión que queremos crear. Se debe seleccionar SSH tal como se muestra en la figura 4.2. Se debe introducir la dirección IP o el nombre del equipo [24] en el que se va a instalar OpenGnsys (se ha optado por esta opción y se ha de poner el nombre FQDN), el usuario con el que se va a acceder al equipo (este usuario lo hemos creado en la instalación de Ubuntu Server que se ha visto en la sección 4.1.1) y el puerto predeterminado para el protocolo SSH (22). Finalmente, se debe presionar sobre el botón «OK».

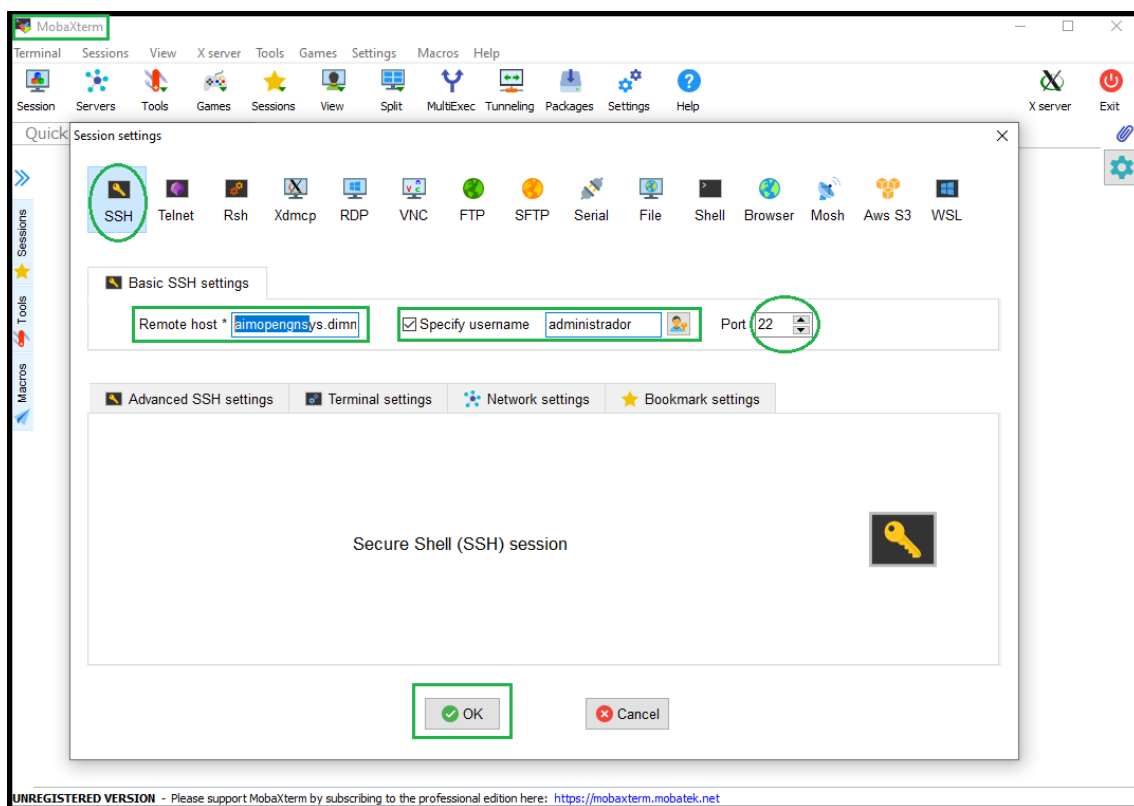


Figura 4.2: Configuración básica conexión SSH

Tal como se observa en la figura 4.3, se ha realizado una conexión SSH al equipo donde se va a instalar OpenGnsys mediante el usuario que se ha creado (administrador). Además se puede observar cuál fue la última conexión y desde qué equipo se ha realizado. En la parte inferior se muestra información del monitor de recursos del equipo al que nos hemos conectado (porcentaje de uso de la CPU, uso de memoria, velocidad de subida/bajada red, etc.).

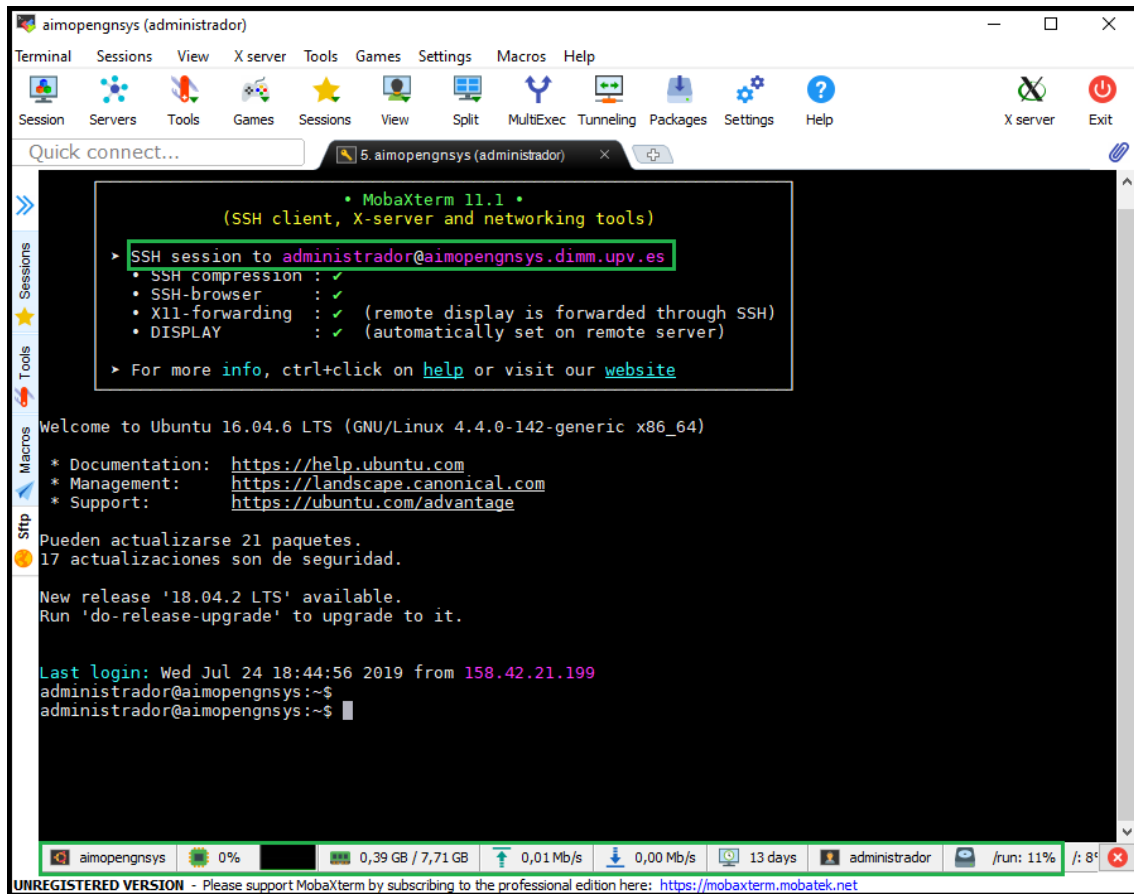


Figura 4.3: Consola SSH al equipo donde se va a instalar el servidor OpenGnsys

4.1.3. Configuración del segundo disco duro

Como se ha visto en la sección 4.1, se dispone de un segundo disco duro de 2 TB de capacidad que va a contener las imágenes de los sistemas operativos de los clientes que queramos clonar. Este disco se ha puesto posteriormente porque no se tenía claro si el servidor del repositorio iba a estar en el mismo equipo que donde se va a instalar OpenGnsys. De hecho es bastante habitual que esté en servidores distintos e, incluso, es frecuente disponer de varios servidores de repositorios para contener las diferentes imágenes de los equipos que queramos clonar. La configuración elegida va en función del número de ordenadores y la distribución geográfica que tengamos.

En nuestro caso, se va a tener un único equipo que tendrá instalado los módulos de servidor OpenGnsys y de servidor del repositorio. Una vez conectado el disco al equipo es necesario configurarlo. Los pasos que se han seguido tras conectarnos al equipo mediante SSH como se ha visto en la figura 4.3 son:

1. Se debe crear una única partición usando todo el tamaño del disco (**/dev/sda**)⁴.

```
sudo fdisk /dev/sda

n (new)
p (primary)
1 (particion)
Enter (principio de tamaño de la particion) – dejar valores por defecto
Enter (fin de tamaño de la particion) – dejar valores por defecto
w (write)
```

2. Se cambia el tipo a la partición a 8e (LVM).

```
sudo fdisk /dev/sda

t (change)
8e (Linux LVM)
w (write)
```

3. Se crea el volumen físico.

```
sudo pvcreate /dev/sda1
```

4. Se crea el grupo de volúmenes og_storage.

```
sudo vgcreate og_storage /dev/sda1
```

5. Se crea el volumen lógico og_images con un tamaño de 1.81 TB mediante el siguiente comando (previamente se puede saber el tamaño del disco mediante el comando `vgdisplay` y hay que tener en cuenta que no podemos usar todo el disco).

```
sudo lvcreate --name og_images --size 1.81TB og_storage
```

6. Se crea el sistema de archivos EXT4 sobre el nuevo volumen lógico y se le asigna la etiqueta IMAGES.

```
sudo mkfs.ext4 -L IMAGES /dev/og_storage/og_images
sudo blkid -L IMAGES
```

7. Se crea el punto de montaje.

```
sudo mkdir -p /opt/opengnsys/images
```

8. Se configura el automontaje. Para ello editamos el archivo `fstab`.

```
sudo nano /etc/fstab
```

Se incluye la siguiente línea de código:

```
LABEL=IMAGES /opt/opengnsys/images ext4 defaults 0 1
```

9. Se reinicia la máquina.

```
sudo reboot
```

⁴Después de introducir la instrucción `fdisk`, se van añadiendo las siguientes instrucciones pulsado la tecla «enter» tras la introducción de cada instrucción

10. Se comprueba el automontaje tal como se puede observar en la figura 4.4.

```
administrador@aimopengnsys:/$  
administrador@aimopengnsys:/$ mount | grep /opt/opengnsys/images  
/dev/mapper/og_storage-og_images on /opt/opengnsys/images type ext4 (rw,relatime,data=ordered)  
administrador@aimopengnsys:/$ █
```

Figura 4.4: Comprobación del automontaje del disco de 2 TB

En este punto se dispone de una distribución Linux en un equipo físico configurado con dos discos duros (el del sistema y el que se va a utilizar para almacenar las imágenes de los sistemas). Se ha de proceder a la instalación de OpenGnsys.

4.2 Instalación de OpenGnsys

El proceso de instalación de OpenGnsys se ha basado principalmente en la documentación del curso en línea que he realizado en el primer cuatrimestre de 2019 a través de la plataforma **docencia-net** [25], más concretamente se hace referencia a [26].

En este punto, ya estamos conectados al equipo donde vamos a instalar OpenGnsys mediante SSH tal como se ha visto con anterioridad. Disponemos del usuario administrador que se creó en la sección 4.1.1. Dicho usuario es capaz de ejecutar instrucciones con privilegios avanzados o *root* mediante el comando *sudo* al principio de la instrucción. Al ejecutar dicha instrucción con elevación de privilegios se nos pide que introduzcamos su contraseña.

El primer comando que se va a lanzar desde la consola SSH, nos va a servir para actualizar la lista de paquetes disponibles y sus versiones de Ubuntu y posteriormente actualizarlos.

```
sudo apt-get update && sudo apt-get upgrade
```

Se necesita descargar las herramientas usadas para descargar OpenGnsys (se distribuye por internet). Para ello se utiliza este comando:

```
sudo apt-get install wget subversion
```

Se puede hacer una rápida comprobación de subversion [27] por medio de este comando:

```
svn info https://opengnsys.es/svn/trunk
```

Se va a proceder a descargar y descomprimir el código de OpenGnsys Server 1.1.0. Para ello se emplea los siguientes comandos:

```
cd /home/administrador  
wget -c http://www.opengnsys.es/trac/downloads/opengnsys-1.1.0-last.tar.gz  
tar xvzf opengnsys-1.1.0-last.tar.gz --directory /tmp/  
cd /tmp/opengnsys/installer  
ls
```

A la hora de lanzar el instalador de OpenGnsys se van a crear una serie de usuarios con sus respectivas contraseñas. Irán apareciendo sus valores por defecto permitiendo su modificación (es importante cambiar tanto los usuarios como las contraseñas por defecto

para evitar el acceso a nuestro servidor por personas ajenas a nuestra organización así como mantenerlas protegidas) tal como se puede observar en la figura 4.5.

```
#!/bin/bash

#####
##### Script instalador OpenGnsys
##### Autor: Luis Guillén <lguillen@unizar.es>
#####

#### AVISO: Puede editar configuración de acceso por defecto.
#### WARNING: Edit default access configuration if you wish.
DEFAULT_MYSQL_ROOT_PASSWORD="passwordroot" # Clave por defecto root de MySQL
DEFAULT_OPENGNSYS_DB_USER="usuog" # Usuario por defecto de acceso a la base de datos
DEFAULT_OPENGNSYS_DB_PASSWD="passusuog" # Clave por defecto de acceso a la base de datos
DEFAULT_OPENGNSYS_CLIENT_PASSWD="og" # Clave por defecto de acceso del cliente
```

Figura 4.5: Primeras líneas del *script* de instalación de OpenGnsys

4.2.1. Versión de ogLive a utilizar

En los equipos clientes (ordenadores de aulas, estaciones de trabajo, equipos de profesores, ordenadores de laboratorios, etc.) se va a cargar un sistema operativo a través de la red vía PXE llamado ogLive.

La versión de OpenGnsys que se va a instalar es la 1.1.0 y se puede tener varias versiones de ogLive de manera simultánea; es decir, se puede instalar una versión de 32 bits de ogLive para los equipos de 32 bits o un ogLive de 64 bits para equipos de 64 bits. Se ha optado por elegir esta última opción por si nos encontráramos con equipos con un hardware más modesto de 32 bits. Cada ogLive necesita realizar una descarga de unos 700 MB y ocupa un espacio de almacenamiento de 1 GB.

4.2.2. Ejecución *script* instalación

Para ejecutar el *script* de instalación de OpenGnsys se debe lanzar desde nuestra conexión SSH (figura 4.3) el siguiente código:

```
sudo bash /tmp/opengnsys/installer/opengnsys_installer.sh
```

En la instalación de OpenGnsys se realizan los siguientes pasos [26]:

- Se solicita confirmación o cambio de los usuarios que se van a crear para OpenGnsys, tal como se vio en la figura 4.5.
- Se solicita la versión del ogLive para los clientes. Se ha optado como se ha explicado en la sección 4.2.1 por instalar la dos versiones (32 y 64 bits).
- Se guarda un histórico de operaciones en la siguiente ruta: /opt/opengnsys/log/opengnsys_installer.log.
- Se realiza en caso de necesidad la descarga de las dependencias del sistema operativo.
- Se crea la estructura de directorios de OpenGnsys en /opt/opengnsys.
- Se realiza una configuración por defecto de los servicios del sistema.
- Se genera la documentación de la interfaz de programación de aplicaciones (API).

- Se instala la imagen del cliente OpenGnsys -ogLive-.
- Se compilan, instalan y arrancan los servicios de OpenGnsys.
- Finalmente, se muestra un resumen de la instalación.

4.2.3. Verificación del proceso de instalación

Llegados a este punto se hace necesario una rápida verificación de que el proceso de instalación de OpenGnsys se ha realizado satisfactoriamente. Para ello un primer paso será verificar que la estructura de directorios creada por el instalador de OpenGnsys es la adecuada. Se lanza el siguiente comando desde nuestra consola SSH.

```
ls -l /opt/opengnsys
```

El resultado que se obtiene es el que se muestra en la figura 4.6.

```
administrador@aimopengnsys:/$ ls -l /opt/opengnsys/
total 40
drwxr-xr-x  2 root root    4096 jun  8 18:10 bin
drwxr-xr-x 10 root root    4096 jun  8 18:10 client
drwxr-xr-x  3 root root    4096 jun  8 18:10 doc
drwxr-xr-x  2 root root    4096 jul  5 20:46 etc
drwxrwxr-x  4 root opengnsys 4096 jul  5 21:07 images
drwxr-xr-x  2 root root    4096 jul  7 14:57 lib
drwxr-xr-x  4 root root    4096 jul 10 06:25 log
drwxr-xr-x  2 root root    4096 jun  8 17:36 -p
drwxr-xr-x  2 root root    4096 jun  8 18:10 sbin
lrwxrwxrwx  1 root root      17 jun  8 18:10 tftpboot -> /var/lib/tftpboot
drwxr-xr-x 20 root root    4096 jun  8 18:12 www
administrador@aimopengnsys:/$
```

Figura 4.6: Estructura de directorios creada con el instalador de OpenGnsys

Mediante el siguiente comando se puede verificar la versión de la aplicación que se ha instalado (versión 1.1.0a).

```
administrador@aimopengnsys:/$ cat /opt/opengnsys/doc/VERSION.txt
OpenGnsys 1.1.0a r20190603.1237a9a
```

Se comprueba los sistemas clientes -ogLives- que ofrece el servidor mediante el siguiente comando (se observa que tenemos dos clientes de 32 y 64 bits respectivamente):

```
administrador@aimopengnsys:~$ sudo /opt/opengnsys/bin/oglivecli show all
{
  "distribution": "xenial",
  "kernel": "4.8.0-39-generic",
  "architecture": "amd64",
  "revision": "r5331",
  "directory": "ogLive-xenial-4.8.0-amd64-r5331",
  "iso": "ogLive-xenial-4.8.0-39-generic-amd64-r5331.iso"
},
{
  "distribution": "precise",
  "kernel": "3.2.0-23-generic",
  "architecture": "i386",
  "revision": "r5159",
  "directory": "ogLive-precise-3.2.0-i386-r5159",
  "iso": "ogLive-precise-3.2.0-23-generic-r5159.iso"
}
```

Se comprueba el sistema ogLive que hay por defecto mediante el siguiente comando:

```
administrador@aimopengnsys:/$ sudo /opt/opengnsys/bin/oglivecli show default
{
  "distribution": "xenial",
  "kernel": "4.8.0-39-generic",
  "architecture": "amd64",
  "revision": "r5331",
  "directory": "ogLive-xenial-4.8.0-amd64-r5331",
  "iso": "ogLive-xenial-4.8.0-39-generic-amd64-r5331.iso"
}
```

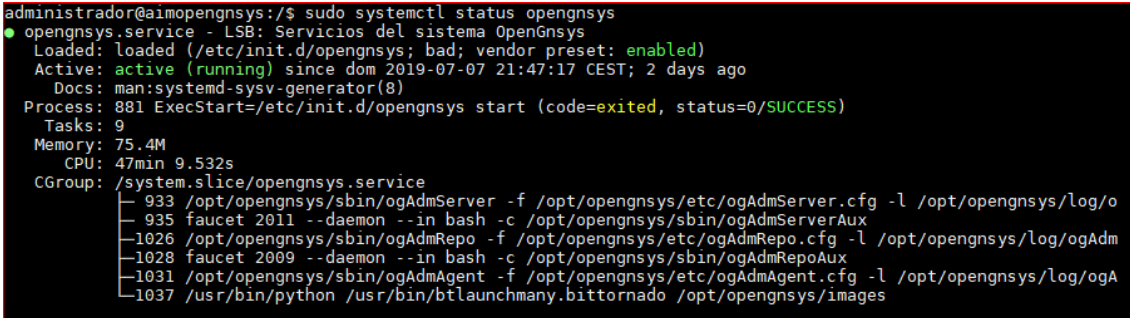
Es conveniente realizar una copia de seguridad del log de la instalación que se ha realizado; para ello se emplea el siguiente comando:

```
sudo cp /opt/opengnsys/log/opengnsys_installer.log /root/
```

Se puede verificar los servicios de OpenGnsys mediante el siguiente comando:

```
sudo systemctl status opengnsys
```

Tal como se observa en la figura 4.7, todo es correcto.



```
administrador@aimopengnsys:/$ sudo systemctl status opengnsys
● opengnsys.service - LSB: Servicios del sistema OpenGnsys
   Loaded: loaded (/etc/init.d/opengnsys; bad; vendor preset: enabled)
   Active: active (running) since dom 2019-07-07 21:47:17 CEST; 2 days ago
     Docs: man:systemd-sysv-generator(8)
  Process: 881 ExecStart=/etc/init.d/opengnsys start (code=exited, status=0/SUCCESS)
    Tasks: 9
   Memory: 75.4M
      CPU: 47min 9.532s
   CGroup: /system.slice/opengnsys.service
           └─ 933 /opt/opengnsys/sbin/ogAdmServer -f /opt/opengnsys/etc/ogAdmServer.cfg -l /opt/opengnsys/log/o
           └─ 935 faucet 2011 --daemon --in bash -c /opt/opengnsys/sbin/ogAdmServerAux
           └─1026 /opt/opengnsys/sbin/ogAdmRepo -f /opt/opengnsys/etc/ogAdmRepo.cfg -l /opt/opengnsys/log/ogAdm
           └─1028 faucet 2009 --daemon --in bash -c /opt/opengnsys/sbin/ogAdmRepoAux
           └─1031 /opt/opengnsys/sbin/ogAdmAgent -f /opt/opengnsys/etc/ogAdmAgent.cfg -l /opt/opengnsys/log/ogA
           └─1037 /usr/bin/python /usr/bin/btlaunchmany.bittornado /opt/opengnsys/images
```

Figura 4.7: Verificación de servicios de OpenGnsys

Por último se puede comprobar que todos los servicios están iniciados. Para ello se lanza el siguiente comando a la consola SSH.

```
sudo systemctl status apache2
sudo systemctl status mysql
sudo systemctl status smb
sudo systemctl status nmbd
sudo systemctl status isc-dhcp-server
sudo systemctl status tftpd-hpa
sudo systemctl status cron
```

En la figura 4.8 se muestra el acceso a la consola web de OpenGnsys. Para ello desde nuestro navegador accedemos a la URL <https://NombreServidor/opengnsys/>.

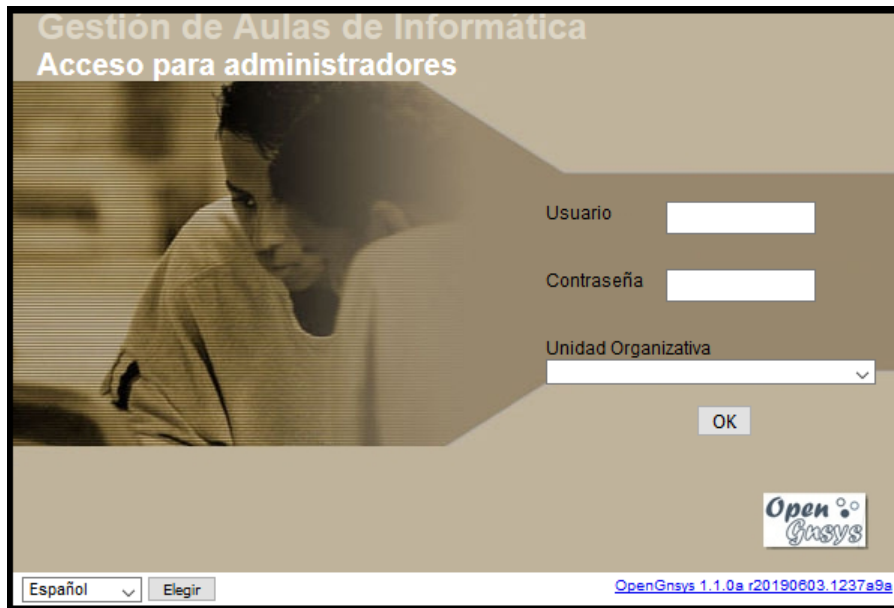


Figura 4.8: Acceso a la consola web de OpenGnsys

4.2.4. Postconfiguración

Como se ha visto en la sección 4.2.2, en la instalación de OpenGnsys se realiza una configuración por defecto para los servicios principales del sistema. Dicha configuración consiste en:

1. Se deshabilita algunos aspectos generales de seguridad relacionado con el cortafuegos del sistema.
2. Configura y activa el servidor OpenGnsys, el servidor del repositorio y el agente del motor de clonación de OpenGnsys.
3. Se configura el servicio DHCP aunque el equipo donde se ha instalado ya estaba dado de alta en la red de la universidad y utiliza el servidor DHCP de la misma.
4. Se genera unas plantillas por defecto para el servicio de arranque por red PXE.
5. Se configuran los servicios de la interfaz de administración web.
6. Se prepara el sistema de archivos Samba [28] utilizado por el repositorio de OpenGnsys.
7. Se genera ficheros auxiliares del agente.

Además, también realiza una configuración por defecto para la interfaz de administración web de OpenGnsys:

8. Se crea un usuario superadministrador y su contraseña (se realiza en la ejecución del *script* de instalación de OpenGnsys).
9. Crea una Universidad (Organización) por defecto formada por una unidad organizativa.
10. Se definen los datos por defecto del servidor y del repositorio.

4.2.4.1. Ficheros de configuración de OpenGnsys

En OpenGnsys para cada componente del sistema se dispone de un fichero de configuración. Este hecho posibilita su distribución en varias máquinas. Dichos ficheros de configuración se pueden localizar en los siguientes ficheros de texto tal como se indica en [26]:

- **Servidor OpenGnsys:** `/opt/opengnsys/etc/ogAdmServer.cfg`
- **Repositorio OpenGnsys:** `/opt/opengnsys/etc/ogAdmRepo.cfg`
- **Agente clonado OpenGnsys:** `/opt/opengnsys/etc/ogAdmAgent.cfg`
- **Cliente OpenGnsys:** `/opt/opengnsys/client/etc/ogAdmClient.cfg`
- **Administrador web OpenGnsys:** `/opt/opengnsys/www/controlacceso.php`
- **Servicios iniciados por defecto:** `/etc/default/opengnsys`

Además se dispone de otros ficheros de configuración:

- **DHCP:** `/etc/dhcp/dhcpd.conf`
- **TFTP/PXE:** `/opt/opengnsys/tftpbppt/menu.lst/*`
- **Samba (repositorio):** `/etc/samba/*`

4.2.5. Instalación de certificado para acceso seguro a la consola Web

Cuando se acaba de instalar el servidor OpenGnsys y se accede a través de un navegador (se ha estado utilizando Mozilla Firefox) a la consola web de OpenGnsys tal como se puede observar en la figura 4.9, se percibe que no se dispone de una conexión segura.

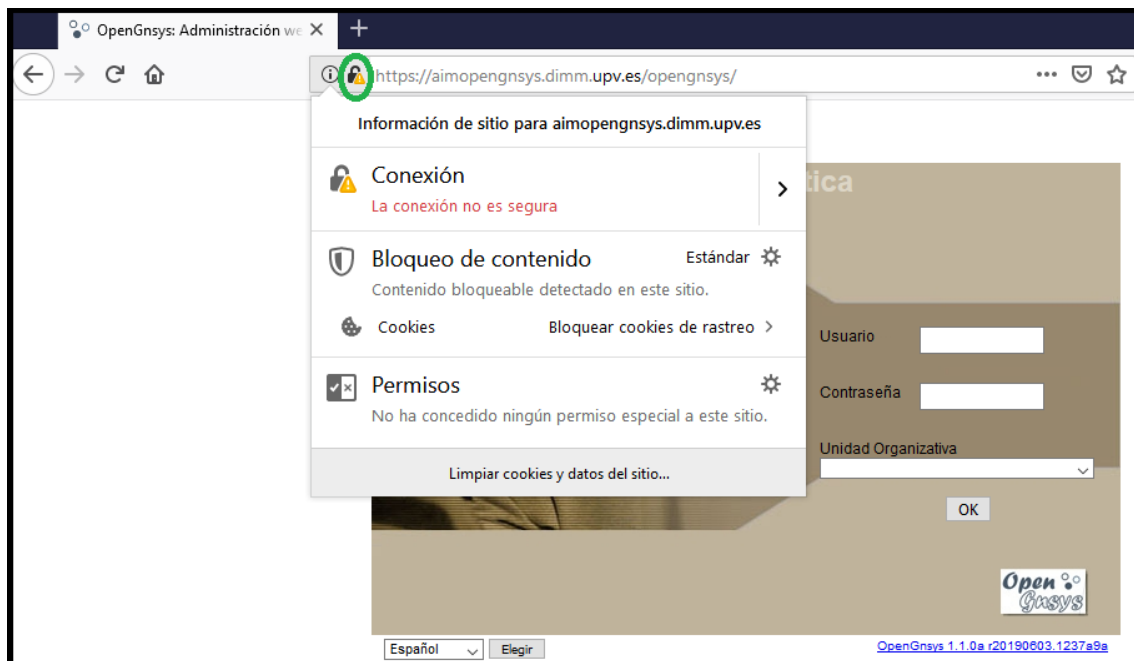


Figura 4.9: Acceso consola web de OpenGnsys tras su instalación

En un primer momento se podría pensar en poner un certificado gratuito SSL para el acceso seguro a la web de OpenGnsys [29]. Nuestra universidad tiene un servicio para solicitar certificados para servidores con lo que se ha optado por solicitar el certificado a la propia universidad. Se han tenido que realizar una serie de pasos:

1. Generar una solicitud para obtener un certificado para el servidor OpenGnsys.
2. Generar el CSR.
Para generarlo se ha utilizado OpenSSL mediante la siguiente orden:

```
openssl req -new -nodes -keyout certificado.key -out certificado.  
csr -config certificado.conf -keyform PEM
```

3. Cuando se complete la solicitud dispondremos del *certificado.key* y del *certificado.pem*.

Una vez se dispone de los archivos necesarios, se ha de proceder a instalar el certificado en el servidor. Antes de ello se añade al fichero *certificado.pem* el contenido del fichero *cadDigiCert.pem* que contiene una cadena con los certificados de todas las Entidades Certificadoras que se necesitan [30]. Para ello se han seguido los pasos siguientes desde la consola SSH:

1. Copiar los ficheros *certificado.pem* y *certificado.key* en el directorio donde se encuentre la configuración de Apache [31].
Se ha copiado los dos archivos en la ruta */etc/apache2/*.
2. Se ha de modificar los archivos de configuración de Apache.
Se ha editado el archivo */etc/apache2/sites-available/default-ssl.conf*, tal como se puede ver en la figura 4.10.
3. Se debe reiniciar el servicio Apache.
Se ha lanzado el siguiente comando:

```
sudo /etc/init.d/apache2 restart
```

```

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
# Se ha comentado estas líneas
#SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
#SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

# --- Modificad| Fede Olmeda
# ModificaciÃn para que coja el certificado que se ha solicitado a digicert
SSLCertificateFile /etc/apache2/certificado.pem
SSLCertificateKeyFile /etc/apache2/certificado.key
# ----- Final modificaciones para certificado

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convenience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

```

Figura 4.10: Modificación configuración Apache para incluir certificado

Una vez modificado el fichero de configuración de Apache y reiniciado el servicio, se procede a entrar de nuevo a la consola de acceso a la web de OpenGnsys. El resultado con el acceso seguro se puede observar en la figura 4.11.

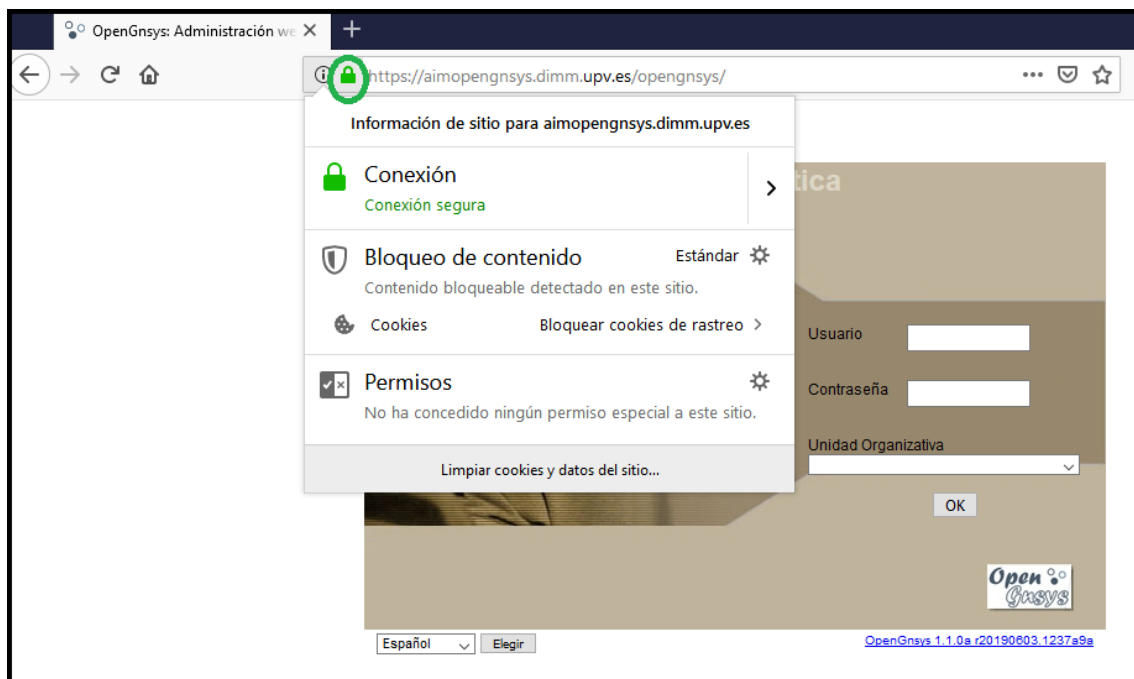


Figura 4.11: Acceso seguro a consola web de OpenGnsys

4.3 Esquema lógico de OpenGnsys

OpenGnsys ofrece una serie de elementos para poder realizar una representación lógica de una organización. Los elementos definidos en OpenGnsys son:

- **Usuarios.**
Pueden haber de dos tipos: superadministrador y administrador. Los superadministradores son los encargados de crear usuarios y unidades organizativas mientras que los usuarios administradores se encargan de la gestión de una unidad organizativa que les ha sido delegada.
- **Grupo de entidades.**
Son contenedores de entidades.
- **Entidades.**
Se definen aquí las unidades organizativas.
- **Unidades organizativas.**
Su principal función es agrupar objetos con fines administrativos y que se permita delegar su autoridad.
- **Grupo de aulas.**
Son contenedores de aulas.
- **Aulas.**
Se definen aquí ordenadores y grupos de ordenadores. Definen un ámbito de aplicación y un ámbito de red de una unidad organizativa.
- **Grupo de ordenadores.**
- **Ordenador.**

El esquema lógico de OpenGnsys permite a los administradores de OpenGnsys:

- **Organizar** los equipos informáticos a gestionar en una estructura de contención jerárquica.
- La **Administración descentralizada** de las unidades organizativas a través de la delegación de responsabilidades administrativas a otros usuarios.

En la figura 4.12 se muestra un esquema lógico con una estructura jerárquica de una organización llamada «Universidad» (se podría tomar como ejemplo a la UPV como organización y los campus serían los que dispone la universidad -Vera, Gandía y Alcoi-. Las entidades serían los edificios que hay en los diferentes campus y las unidades organizativas serían las escuelas, facultades y departamentos). Se realiza el esquema lógico de una organización siguiendo algún criterio: geográfico, departamental o funcional. A pesar de estar orientado a la estructura de una universidad se podría aplicar a una empresa -pequeña, mediana o grande- sin ningún problema.

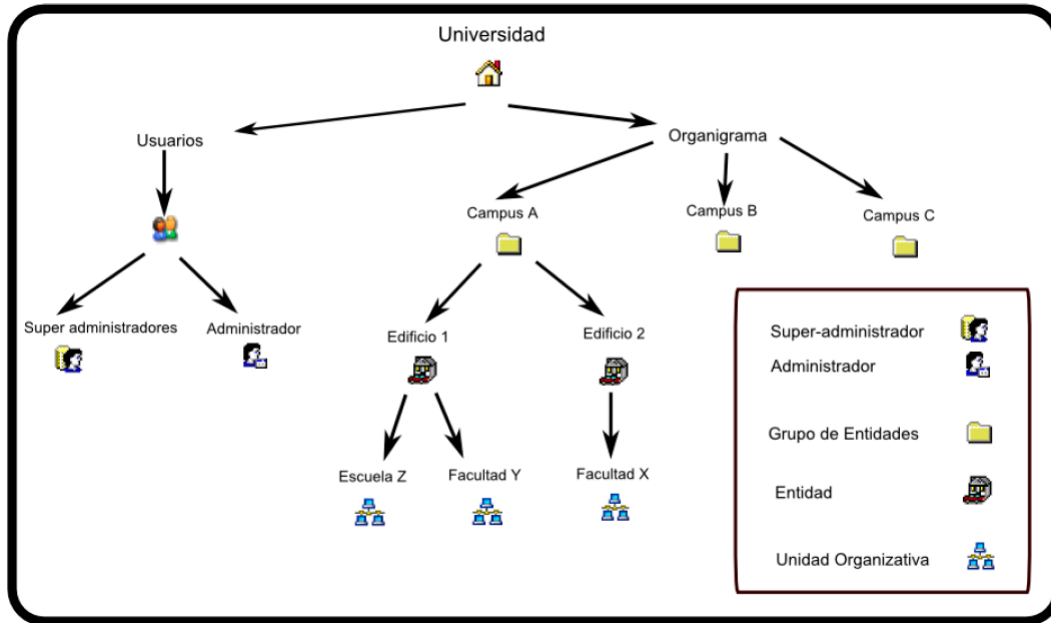


Figura 4.12: Ejemplo de esquema lógico de una organización con OpenGnsys

Como se puede observar en la figura 4.13 se tiene una configuración de una organización con una estructura descentralizada. En la parte superior se observa cómo un usuario superadministrador entra en la consola y genera un esquema lógico jerárquico de la organización (usuarios, grupos de entidades, entidades y unidades organizativas). Es necesario crear una serie de usuarios administradores para poder delegar a ellos la gestión de las unidades organizativas. En la parte inferior se puede observar que para gestionar una unidad organizativa se ha de entrar en la consola con las credenciales (usuario y contraseña) del usuario al que han delegado su gestión. Una vez entrada en la consola para gestionar una determinada unidad organizativa, se va a poder crear un esquema lógico por medio de los elementos de grupos de aulas, aulas, grupos de ordenadores y ordenadores en función de la configuración que necesite la organización a gestionar. Cada elemento mencionado, al ser un esquema jerárquico, define un ámbito de aplicación que limita o define el ámbito de aplicación de una tarea sobre un determinado agrupamiento de equipos informáticos a gestionar [32].

El administrador de una unidad organizativa puede a su vez delegar tareas específicas que son accesibles por el usuario desde el propio ordenador gestionado (mediante el cliente ogLive). Estas tareas delegadas son incluidas en un menú que puede requerir de autenticación -menú privado- o no -menú público-.

Con la instalación por defecto de OpenGnsys se dispone de un usuario superadministrador y se configura de manera automática un esquema que ofrece una estructura básica para la gestión de una organización de tipo pequeña. El nombre de la organización se define como «Universidad» y se dispone de una única unidad organizativa que se ha definido dentro de una entidad, identificada como «Entidad default».

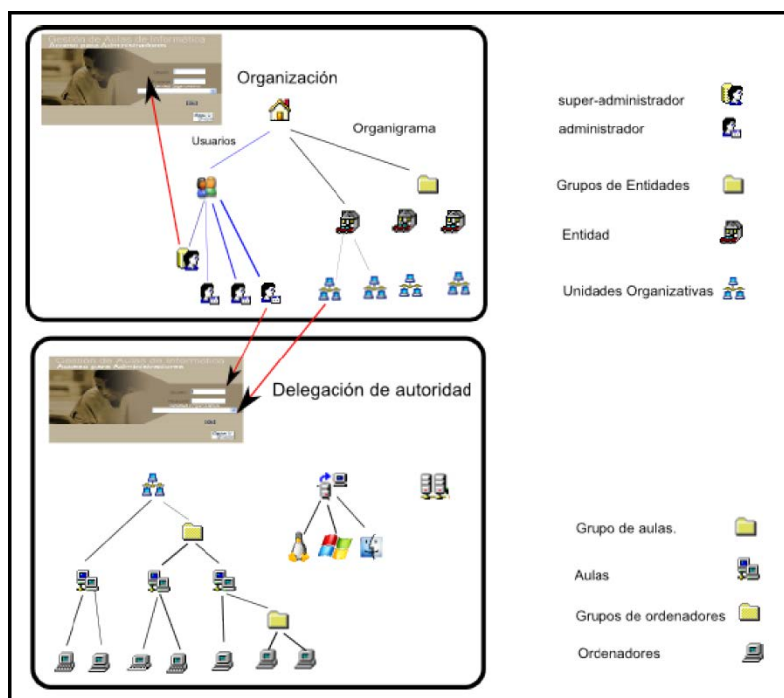


Figura 4.13: Administración descentralizada de una organización con OpenGnsys

4.4 Herramienta de administración de OpenGnsys

OpenGnsys proporciona unas herramientas necesarias para su administración. Dichas herramientas deben ser gestionadas por usuarios administradores.

El acceso a la consola web se realiza desde el navegador web Firefox (se pueden utilizar otros navegadores pero hay determinadas operaciones solo funcionan correctamente desde Firefox) usando la dirección URL <https://NombreServidor/opengnsys/> tal como se ve en la figura 4.8.

En función de los derechos de administración que posee un usuario, podemos tener dos tipos de vistas diferentes:

- **Acceso a la vista para el administrador de la consola.**

Se realiza mediante la autenticación de un usuario superadministrador y sin seleccionar ninguna unidad organizativa a gestionar. El resultado se puede observar en la figura 4.14.

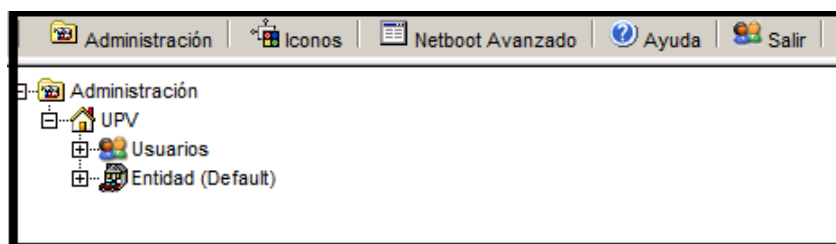


Figura 4.14: Acceso a la vista para el administrador de la consola

- **Acceso a la vista para el administrador de una unidad organizativa de OpenGnsys.**

Se realiza mediante la autenticación de un usuario administrador y seleccionando la unidad organizativa con derechos de administración. El resultado se puede observar en la figura 4.15.

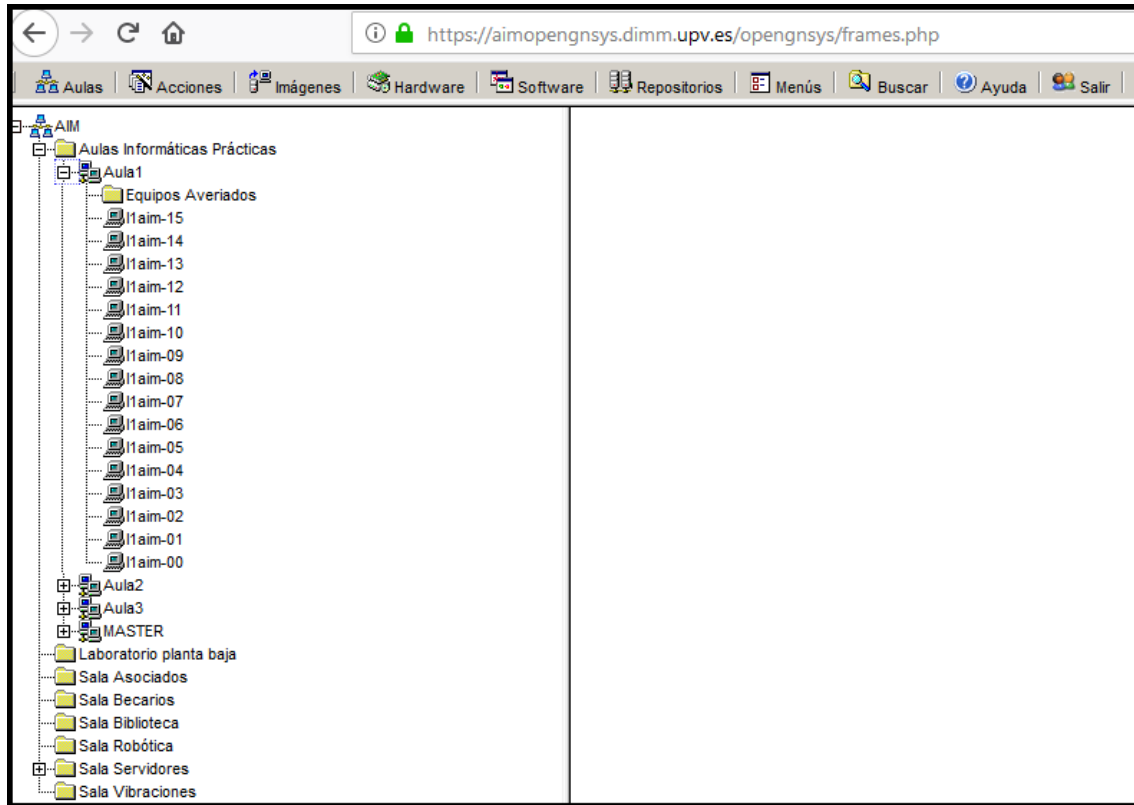


Figura 4.15: Acceso a la vista para el administrador de una OU de OpenGnsys

En cualquiera de las dos vista que se utilicen a través de la consola web se dispone de un área de trabajo que está dividida en tres zonas que tienen la misma función, tal como se puede observar en la figura 4.16:

- Menú de herramientas (1).
Está situada en la parte superior de la consola.
- Área del árbol de objetos (2) .
Está situada en la parte de la izquierda y la mayoría de sus elementos disponen de un menú contextual.
- Área de operaciones y resultados (3).
Situado en la parte de la derecha y en ella aparecen los formularios de operación o los resultados obtenidos sobre un elemento del árbol de objetos.



Figura 4.16: Área de trabajo de OpenGnsys

Si seleccionamos alguna opción del área superior o menú de herramientas, se nos va a mostrar en el área de la izquierda (árbol de objetos) todos aquellos elementos sobre los cuales podemos operar.

Si se pulsa el botón derecho sobre un objeto del árbol de objetos aparecerá su **menú contextual** con todas las opciones a realizar sobre dicho objeto. Cualquier opción que se elija del menú contextual mostrará en el área de la derecha los detalles de la opción elegida.


Si realizamos un cambio de valor sobre un formulario de operación que tenemos en el área de la derecha se modificará automáticamente sobre el objeto sobre el que actúa pero hay ocasiones hay que pulsar «F5» para actualizar el navegador y visualizar dichos cambios en la consola.

4.5 Consola de administración

Como se ha visto en la sección 4.4, se dispone de una vista para la administración de la consola web de nuestra organización. Se entra con un usuario superadministrador y sin seleccionar ninguna unidad organizativa a gestionar, tal como se puede ver en la figura 4.16.

La herramienta identificada como «Administración» es la más importante de la consola de administración. Desde aquí se gestionan los usuarios, las unidades organizativas y los derechos de acceso.

Los objetos disponibles en la herramienta «Administración» son:

 Es el contenedor de tipo **universidad** que constituye el nivel superior en la estructura de OpenGnsys y representa a la organización.




Es el contenedor de tipo usuarios. Desde aquí se gestionan los usuarios.



Es el contenedor entidad donde están las unidades organizativas (se gestiona desde aquí).




Es el contenedor de unidad organizativa (se usa para agrupar objetos con fines administrativos con la posibilidad de delegar su administración).

Lo primero que se ha realizado es cambiar el nombre al contenedor de tipo universidad. Para ello y desde el menú contextual «propiedades» sobre el objeto  aparece el formulario que se puede observar en la figura 4.17. Se ha puesto como nombre: «UPV» y en la parte de comentarios se ha dejado como estaba inicialmente.



Gestión Universidades Modificar	
Nombre	UPV
Comentarios	Esta Universidad se crea automáticamente en el proceso de instalación de OpenGnsys
<input type="button" value="Cancel"/> <input type="button" value="Accept"/>	


Figura 4.17: Gestión Universidades

En el menú contextual «Variables de entorno» sobre el objeto  aparece en el área de operaciones un formulario donde muestra la información que vemos en la figura 4.18. Esta opción se podría utilizar en el caso de querer cambiar el servidor de OpenGnsys para gestionar la organización.



Gestión de Variables de Entorno Modificar	
IP Servidor de administración	158.42.207.47
Puerto Servidor de administración	2008
Método de clonación preferido	UNICAST
<input type="button" value="Cancel"/> <input type="button" value="Accept"/>	

Figura 4.18: Gestión de variables de entorno

Mediante el menú contextual sobre el objeto  se puede crear un nuevo grupo de entidades (aparecería un formulario donde se solicita el nombre del grupo y algún comentario del mismo) o se podría crear una nueva entidad (solicitando el nombre de la entidad y un comentario).


La herramienta identificada como «iconos» permite la creación, modificación y eliminación de una serie de iconos que se van a utilizar en la aplicación OpenGnsys. No se va a entrar en detalle y se deja la configuración por defecto.

La herramienta identificada como «Netboot Avanzado» permite la gestión del arranque avanzado de los clientes ogLive. Tampoco se va a modificar nada de la configuración que viene por defecto.


La herramienta identificada como «Ayuda» muestra el manual de usuario de la aplicación, la documentación de la API de OpenGnsys, la configuración de los clientes, los cambios en la versión del OpenGnsys que tenemos instalada, la lista de correos de usuarios de la aplicación, la web del proyecto y el acerca de. Desde aquí podemos acceder a muchísima información del proyecto de una manera sencilla.

Por último, la herramienta identificada como «Salir» nos permite cerrar la sesión actual y aparece la página de acceso a la aplicación como se observa en la figura 4.8.

4.5.1. Gestión de usuarios


Para dar de alta un usuario se utiliza el menú contextual del objeto usuarios (). Mediante el menú contextual del usuario que queremos gestionar, se puede modificar y eliminar dicho usuario.

- **Crear nuevo usuario superadministrador.**

Se ha de seleccionar el menú contextual del objeto  «Nuevo Superadministrador». Aparecerá en el área de operaciones un formulario para insertar un nuevo usuario. Se ha rellenado dicho formulario tal como aparece en la figura 4.19.



Gestión Usuarios (SuperAdministrador)
Insertar

Usuario	AdministradorAIM 
Password*
Confirmar password
Nombre completo	Administrador OpenGnsys AIM
E-mail	
Idioma	Español 
API key	La API key se generará automáticamente al insertar el usuario.




 

Figura 4.19: Creación usuario superadministrador

- **Crear nuevo usuario administrador.**

Se ha de seleccionar el menú contextual del objeto  «Nuevo Administrador». Se ha rellenado el formulario que aparece en el área de operaciones tal como aparece en la figura 4.20.

Gestión Usuarios (Administrador)	
Insertar	
Usuario	GestorAIM
Password*
Confirmar password
Nombre completo	Gestor UO AIM
E-mail	
Idioma	Español
API key	La API key se generará automáticamente al insertar el usuario.
<input type="button" value="Cancel"/> <input type="button" value="Accept"/>	

Figura 4.20: Creación usuario administrador

Una vez creados los usuarios (administrador y superadministrador) las opciones del menú contextual sobre uno de los usuarios creados son: **Asignar unidades organizativas** (esta opción asigna el usuario seleccionado a una unidad organizativa - figura 4.21-), **propiedades** (permite modificar datos del usuario) o **eliminar** (borra el usuario previa confirmación que se desea su eliminación).

Gestor UO AIM

Asignar centros

Centros asignados

T	Nombre del centro
Centros disponibles	
T	Nombre del centro
<input type="checkbox"/>	AIM

(*)Para asignar algún centro a un administrador determinado debe hacer click sobre la casilla de verificación correspondiente.

Figura 4.21: Asignar usuario a una unidad organizativa

4.5.2. Gestión de unidades organizativas

Para crear una unidad organizativa se realiza desde el menú contextual «Nueva unidad organizativa» de un objeto de tipo entidad (). Las unidades organizativas jerárquicamente dependen de un objeto de tipo entidad. Aparece el formulario que se observa en la figura 4.22 para su creación. Se deben rellenar los valores necesarios y presionar al botón de aceptar para confirmar la creación de la nueva unidad organizativa.

The screenshot shows a web form titled "Gestión de Unidades Organizativas" with a sub-header "Insertar". The form contains three input fields: "Nombre de la unidad" (empty), "Comentarios" (empty), and "Directorio" (empty). Below the fields is a grey box with the text: "*Si se elige separar por unidades organizativas, el directorio /opt/opengnsys/images debe contener un subdirectorio llamado como el directorio indicado." At the bottom are two buttons: "Cancel" with a red 'X' icon and "Accept" with a green checkmark icon.

Figura 4.22: Creación de una nueva unidad organizativa

Mediante la opción del menú contextual «Propiedades» se puede modificar los valores asociados a la unidad organizativa sobre la que se actúa tal como se puede ver en la figura 4.23 y permite sus cambios.

El campo directorio permite configurar el sistema para aislar completamente las imágenes de los sistemas operativos entre las diferentes unidades organizativas. En nuestro caso se deja en blanco porque únicamente tenemos una unidad organizativa.

The screenshot shows the same web form as in Figure 4.22, but in "Modificar" mode. The "Nombre de la unidad" field contains the text "AIM". The "Comentarios" field contains the text "Unidad Organizativa del AIM (Edificio 5E)". The "Directorio" field is empty. The same grey box with the warning text and the "Cancel" and "Accept" buttons are present at the bottom.

Figura 4.23: Propiedades sobre una unidad organizativa

Mediante la opción del menú contextual «Asignar Administradores» se puede seleccionar el usuario que va a gestionar dicha unidad organizativa tal como se puede ver en la figura 4.24. Aparece el listado de todos los usuarios disponibles y de aquellos que tienen derecho de administración. También se puede revocar el derecho de autoridad a un administrador y asignar el usuario que va a gestionar la unidad organizativa desde las opciones de usuario tal como se ha visto en la figura 4.21.



Figura 4.24: Asignación de un usuario administrador sobre una unidad organizativa

Mediante la opción del menú contextual «Eliminar» se puede eliminar la unidad organizativa sobre la que se está actuando. Se ha de confirmar que realmente se desea eliminar la unidad organizativa.

Como nuestra organización es básica solo se dispone de una única unidad organizativa -AIM- (se le ha asignado el usuario superadministrador que se creó en la instalación del OpenGnsys y también el usuario Gestor UO AIM porque era necesario para un apartado que se tratará posteriormente).

4.6 Consola de administración web de una unidad organizativa

Para la realización de la presente sección se ha basado en la documentación del curso que he mencionado anteriormente, en concreto se basa en la documentación referenciada en [33].

La consola de administración web proporciona una serie de herramientas para gestionar una unidad organizativa. Para el acceso a dicha consola se ha de escribir en la barra de direcciones del navegador: <https://NombreServidor/opengnsys/>. Se ha de indicar las credenciales del usuario que gestiona la unidad organizativa y también seleccionar la unidad organizativa sobre la que se quiere trabajar en la pantalla de acceso a la consola de OpenGnsys que se observa en la figura 4.8.

Al entrar en nuestra unidad organizativa «AIM» nos encontramos con la consola de administración que se puede observar en la figura 4.25. Al igual que se trató en la sección 4.4, el área de trabajo de la consola de administración está dividida en tres zonas: la parte superior donde se encuentra la barra de herramientas, la parte izquierda donde se encuentra el árbol de objetos y la parte derecha donde aparecen los formularios de operación o los resultados obtenidos sobre los elementos del árbol de objetos.



Figura 4.25: Consola de administración web de la unidad organizativa AIM

4.6.1. Herramientas de gestión

En la parte superior de la consola de administración web se proporciona una herramienta para gestionar una unidad organizativa. Esta barra de herramientas se puede observar en la figura 4.26.

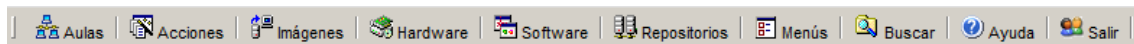


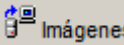
Figura 4.26: Barra de herramientas para la gestión de una unidad organizativa

 **Aulas** Herramienta Aulas.

Permite añadir los objetos del tipo ordenadores, agruparlos en ámbitos de aplicación y realizar operaciones básicas sobre ellos.

 **Acciones** Herramienta Acciones.

Permite gestionar las acciones -procedimientos y tareas (operaciones definidas por el usuario)- que podrán ser programadas o aplicadas a los objetos ordenadores.

 **Imágenes** Herramienta Imágenes.

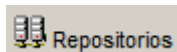
Gestiona la información de los objetos imágenes.

 **Hardware** Herramienta Hardware.

Gestiona el inventario hardware de la unidad organizativa.

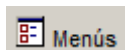
 **Software** Herramienta Software.

Gestiona el inventario software de las imágenes que hayamos creado.



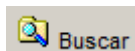
Herramienta Repositorios.

Gestiona los repositorios o contenedores de imágenes asignados a la unidad organizativa. En la instalación de OpenGnsys se crea automáticamente un repositorio «default» para la unidad organizativa por defecto. Si se desea crear una unidad organizativa no se asocia automáticamente ningún repositorio. Lo que se tiene que hacer en ese caso, es dar de alta un nuevo repositorio para asignarlo a esta nueva unidad organizativa.



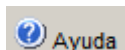
Herramienta Menús.

Gestiona la delegación de determinadas operaciones y/o procedimientos a los usuarios a través de menús asociados a los ordenadores que se muestran por pantalla. Estos menús pueden ser privados o públicos.



Herramienta Buscar.

Permite realizar búsquedas de ordenadores basada en alguna de sus propiedades: nombre, dirección IP, dirección MAC o duplicados.



Herramienta Ayuda.

Se puede acceder al manual de usuario, la documentación de la API que dispone OpenGnsys, la configuración de los clientes, los cambios en la versión, la lista de correos de usuario, información a la web del proyecto y el acerca de OpenGnsys.



Herramienta Salir.

Permite salir de la consola web de la unidad organizativa.

4.6.2. Gestión del repositorio

Cuando se instala un nuevo servidor de OpenGnsys se instalan todos sus componentes: la consola web, la base de datos y también un repositorio. Este repositorio por defecto se ha renombrado como «AIM_REPO» (como ya se comentó con anterioridad, nuestra organización es bastante sencilla y no es necesario disponer de varios repositorios).

El repositorio realiza varias funciones tal como se describe en [33]:

- Contenedor de imágenes de los sistemas de archivos a distribuir entre los distintos ordenadores.
- Gestor de los protocolos de red usados en la restauración de imágenes: *unicast*, *multicast* y *P2P*.

La aplicación web de OpenGnsys necesita para gestionar las operaciones de clonación y particionado:

- Que exista una vinculación entre los objetos de tipo ordenador al objeto repositorio.

- Que exista una vinculación entre los objetos de tipo imagen a un repositorio.

Un repositorio puede pertenecer a varias unidades organizativas, pero con limitaciones:

- Los objetos imágenes no se comparten entre las unidades organizativas.
- Las imágenes creadas por una OU pueden quedar inaccesibles, ya sea porque desde la otra OU se borren o se sobrescriban.

Si se quisiera disponer de varios repositorios sobre una misma OU, deben tener la misma contraseña samba [28] para que el sistema operativo PXE ogLive pueda conectar con uno u otro.

Únicamente en el proceso de creación de una imagen desde un objeto ordenador es necesario que dicho objeto ordenador esté vinculado con un objeto repositorio (que va a contener dicha imagen). En el caso de realizar una restauración de una imagen del sistema sobre un equipo, es el propio sistema operativo PXE ogLive el que gestiona la conexión y vinculación temporal entre un uno u otro repositorio.

Para dar de alta, modificar o eliminar un objeto de tipo repositorio dentro de una OU se realiza con el botón «Repositorios» en la barra de herramientas (figura 4.26).

Para dar de alta un objeto repositorio se ha de realizar desde el menú contextual del objeto principal «Repositorios» pulsando el botón derecho del ratón. Se elige la opción «Añadir Repositorio» y se muestra el formulario de los datos requeridos para su alta tal como se puede observar en la figura 4.27. Se deben rellenar los campos vacíos de dicho formulario. En nuestro caso no se dio de alta un nuevo repositorio sino que se modificó el nombre del repositorio que se crea en la instalación de OpenGnsys y se revisaron que los datos eran correctos.

Gestión Repositorios	
Insertar	
Nombre del Repositorio	<input type="text"/>
Dirección IP	<input type="text"/>
Puerto	2002
Api Token	<input type="text"/>
Comentarios	<input type="text"/>

Ordenadores: 0

Figura 4.27: Alta de un nuevo repositorio

En la figura 4.28 se muestra las propiedades del único repositorio que se dispone. Se observa una serie de información que se puede modificar (nombre del repositorio, dirección IP del servidor del repositorio, el puerto que se utiliza -por defecto es el 2002-, la Api Token -tiene que coincidir con el que aparece en el fichero ogAdmRepo.cfg cuya ruta se encuentra en el apartado 4.2.4.1- y los comentarios para indicar una nota o aclaración

personal). Además nos da información del espacio del repositorio así como de lo que ocupan las imágenes que hay en el mismo. Pasa sacar las propiedades del repositorio se debe elegir el menú contextual «Propiedades» sobre el objeto «AIM_REPO».

Gestión Repositorios
Modificar

Nombre del Repositorio	AIM_REPO	 Ordenadores: 66	
Dirección IP	158.42.207.47		
Puerto	2002		
Api Token	a074e2dcf40023f124b00fe0dfd14b8b		
Comentarios	Repositorio del AIM para contener las imágenes de los sistemas		
Espacio TOTAL	Espacio Ocupado	Espacio Libre	% Ocupado
1.78 TB	260.15 GB	1.53 TB	15 %
Contenido /opt/opensys/images			
Imagen (Tipo)	Tamaño	Modificado	Permisos
W10Base12062019 (img)	7.89 GB	2019-07-02 19:21:58	0644
ImagenFullAIM2019 (img)	71.96 GB	2019-07-05 20:45:47	0644

Cancel Accept

Figura 4.28: Propiedades del repositorio de la OU AIM

Sin acceso a información del repositorio
Revise la conexión con el repositorio
Revise la clave de acceso (campo Api Token debe coincidir con su valor en fichero "ogAdmRepo.cfg")

Figura 4.29: Mensaje de error propiedades repositorio

Hay veces que no se muestra la información sobre el espacio del contenedor y de las imágenes que alberga. En estos casos, se muestre el mensaje de la figura 4.29 y se debe verificar lo siguiente para resolver el problema:

- Que el repositorio esté encendido y accesible por red su dirección IP.
- Que el valor Api Token coincide con el que aparece en `/opt/opensys/etc/ogAdmRepo.cfg`.
- Que el repositorio es un OpenGnsys 1.1.0 o superior.

En la figura 4.30 se muestra la información del repositorio de la unidad organizativa AIM: comentario del repositorio, los ordenadores asignados a dicho repositorio y las imágenes que se alojan en dicho repositorio. Para acceder a dicha información se debe elegir el menú contextual «Información Repositorio» sobre el objeto «AIM_REPO».



Figura 4.30: Información del repositorio de la OU AIM

Por último, se puede eliminar un repositorio mediante el menú contextual «Eliminar Repositorio» sobre el objeto repositorio al que se desea eliminar. Si se acepta la confirmación se elimina dicho repositorio. Si se borra un objeto de tipo repositorio se elimina la referencia de dicho repositorio en la aplicación web de OpenGnsys pero no se elimina ni el servidor de repositorio físico ni los ficheros de imagen que contienen.

4.6.3. Organización de aulas

Mediante la organización de las aulas siguiendo una estructura jerárquica se crean ámbitos de aplicación que delimitan los ordenadores sobre los cuales se ejecutarán las operaciones que se necesiten realizar en un momento determinado.

La estructura típica de organización con OpenGnsys incluye:

- Unidad Organizativa.
- Grupo de aulas.
- Aulas.
- Grupo de ordenadores.
- Ordenadores.

En la figura 4.31 se muestra la jerarquía de aulas de nuestra OU llamada AIM en base al tipo de aulas y la ubicación en el edificio.

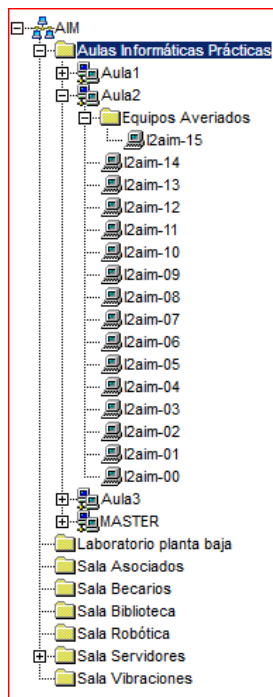


Figura 4.31: Jerarquía de aulas de la OU AIM

4.6.3.1. Gestión de grupos de aulas

La gestión de grupos de aulas se pueden crear y eliminar desde el menú contextual de la OU. Se utilizan para definir un ámbito de aplicación. Se observa en la figura 4.31 que «Aulas Informáticas Prácticas» es un grupo que tiene en su interior una serie de aulas informáticas.

4.6.3.2. Gestión de grupo de ordenadores

La gestión de grupos de ordenadores se crean y eliminan desde el menú contextual de un objeto de tipo aula. Su objetivo es crear un ámbito de aplicación. Se observa en la figura 4.31 que «Equipos Averiadados» es un grupo de ordenadores dentro de un aula -Aula2-.

Un grupo de ordenadores está formado por un conjunto de equipos que normalmente son iguales (comparten un mismo perfil de hardware) o que están en la misma sala, y que comparten una misma distribución de instalación y unos datos de postconfiguración.

4.6.3.3. Gestión de aulas

Mediante un objeto aula se define un ámbito de aplicación y un ámbito de red (define parámetros de red común a todos los objetos de tipo ordenador que dependan de él así como define los datos de los protocolos de red a utilizar en la transferencia de las imágenes de los sistemas operativos -unicast, multicast o P2P-) [33].

En un objeto de tipo aula se puede crear ordenadores y grupos de ordenadores.

Para dar de alta una nueva aula se puede crear desde el menú contextual de la unidad organizativa o de un grupo de aulas, utilizando la opción «Añadir nueva aula». Para eliminar un aula se debe realizar desde el menú contextual del aula, utilizando la opción «Eliminar aula» e implica la eliminación de los equipos que estén en su interior. No es posible mover las aulas entre diferentes grupos de aulas.

Para dar de alta una nueva aula o para modificar las propiedades de un aula ya creada se utiliza un formulario con dos funciones bien diferentes:

- Asignar las propiedades del objeto aula.

En la figura 4.32 se muestra las propiedades específicas de un aula que se quiere dar de alta, alguna de ellas con los valores por defecto.

Asignar propiedades al objeto Aula	
Nombre	<input type="text"/>
Ubicación	<input type="text"/>
Proyector	<input type="checkbox"/>
Pizarra	<input type="checkbox"/>
Aforo	<input type="text" value="0"/>
Foto	<input type="text" value="aula.jpg"/> Ver fotos
Comentarios	<input type="text"/>
Acceso remoto	<input type="checkbox"/> AVISO: activar calendario de eventos de la BD para habilitar esta propiedad
Dirección gateway del aula	<input type="text"/>
Máscara de red del aula	<input type="text"/>
IP Servidor NTP (opcional)	<input type="text"/> (zona horaria por defecto: Europe/Madrid)
IP Servidor DNS (opcional)	<input type="text"/>
URL Servidor Proxy (opcional)	<input type="text"/>
Modo P2P -semillero durante la descarga-	<input type="text" value="peer"/>
Tiempo P2P -semillero después de descarga-	<input type="text"/>
Modo Multicast	<input type="text" value="Full-Duplex"/>
IP Multicast	<input type="text"/>
Puerto Multicast	<input type="text" value="9000"/>
Velocidad Multicast	<input type="text"/>

Figura 4.32: Formulario de propiedades específicas de un aula

- Modificador masivo de propiedades de los objetos de tipo ordenador que dependan jerárquicamente del aula.

En la figura 4.33 se muestra las propiedades que se pueden modificar para todos los ordenadores que estén vinculados a esa aula.

Modificador masivo de propiedades de ordenadores	
Menú	<input type="text"/>
Repositorio	<input type="text"/>
ogLive	<input type="text"/>
Autoexec	<input type="text"/>
Perfil hardware	<input type="text"/>
Validación	<input type="text" value="No"/>
Página login	<input type="text"/>
Página validación	<input type="text"/>

Figura 4.33: Formulario donde se modifican masivamente las propiedades de un aula

Las propiedades que definen un objeto de tipo aula se pueden agrupar en datos de aula física, datos de RemotePC (opcional), datos de red (obligatorio), datos de servicios de red - NTP, DNS, proxy [34] - para el ogLive (opcional) y datos de los protocolos de red P2P y multicast (obligatorio).

Cabe destacar que la propiedad de red determina la importancia del objeto tipo aula, ya que estos datos se propagan a todos los objetos y elementos que dependan jerárquicamente de él y que tienen las siguientes funciones:

- Los parámetros de red permiten que el ogLive se inicie.
- Los parámetros de protocolos de red P2P y multicast permiten que se puedan iniciar los procesos de restauración de las imágenes.
- Habilitar el uso de RemotePC.

Para la introducción de los datos de aula física a la hora de crear un aula se puede ver en la figura 4.34 y en la tabla 4.4 la información que se necesita introducir.

Figura 4.34: Datos de aula física

Campo	Obligatorio	Descripción
Nombre	Sí	Nombre del aula
Ubicación	No	Ubicación del aula (campus, edificio, ...)
Proyector	No	Indica que el aula tiene proyector
Pizarra	No	Indica que el aula tiene pizarra
Aforo	Sí	Es el número máximo de ordenadores que se podrán gestionar
Foto	No	Permite seleccionar la imagen que aparece a la derecha Permite visualizar las imágenes que tenemos disponibles para asignar. Presionar "Ver fotos"
Comentarios	No	Información de interés

Tabla 4.4: Datos de aula física

Respecto a los datos de proyecto RemotePC ([35] y [36]), es opcional. Este proyecto consiste en permitir acceder de forma remota a equipos de aulas TIC de la universidad fuera del horario lectivo desde cualquier lugar y utilizando cualquier dispositivo gracias a las funcionalidades de UDS Enterprise y OpenGnsys. Se hablará con un poco más de detalle más adelante en la sección 4.6.3.4.

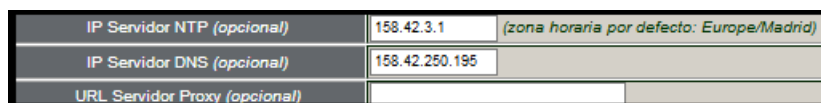
Los datos de red son obligatorios y muy importantes para el correcto funcionamiento del sistema. Está formado por los siguientes datos:

- Dirección IP del router del aula.
En nuestro caso 158.42.207.250.
- Máscara de red del aula.
En nuestro caso 255.255.255.0.

Si estos datos no son correctos, el sistema operativo PXE ogLive no se inicia correctamente dando errores de kernel. Es por ello que se ha de estar seguro de los valores que se ponen para asegurar la estabilidad de estos servicios.

Los datos de servicios de red (NTP, DNS, proxy) para el ogLive son los que se puede ver en la figura 4.35 (opcionales ya que solo se aplican para complementar la configuración de red del sistema operativo PXE ogLive y no son requisito para las tareas básicas de OpenGnsys) son:

- IP servidor NTP (opcional).
Se ha puesto el servidor NTP de la universidad pero se podría poner cualquier otro o dejar vacío.
- IP servidor DNS (opcional).
También se ha puesto el de la universidad pero se podría haber puesto otro o dejado vacío el campo.
- URL servidor proxy (opcional).
En nuestro caso no usamos proxy. El formato que se ha de utilizar es: `http://nombredns:puerto`.



IP Servidor NTP (opcional)	158.42.3.1	(zona horaria por defecto: Europe/Madrid)
IP Servidor DNS (opcional)	158.42.250.195	
URL Servidor Proxy (opcional)		

Figura 4.35: Datos de servicios de red para el ogLive

Los datos de los protocolos de red P2P y multicast son obligatorios. Para el caso del protocolo P2P:

- Modo P2P: es el modo de comportamiento del cliente mientras está descargando la imagen.
Los valores posibles son: *Leecher* -el cliente no comparte mientras descarga la imagen, *Peer* -el cliente, mientras descarga la imagen, comparte los datos que tenga descargados- o *Seeder* -modo de trabajo distribuido en el que el cliente máster se encarga de servir la imagen-.
- Tiempo P2P en segundos. Es el tiempo a compartir la imagen desde que se haya descargado completamente en la caché.

Para nuestro caso se ha elegido la opción «peer» y 60 segundos de semilla para el P2P.

Los datos del protocolo multicast son:

- Modo multicast.
half duplex: transmite y recibe en ambas direcciones pero solo ocurre una transmisión a la vez.
full duplex: transmite y recibe en ambas direcciones al mismo tiempo.

- IP multicast.
La dirección multicast para el aula tendrá el formato **239.255.x.y**, donde la «x» será la VLAN que usemos y la «y» puede ser un valor distinto para cada aula. Es necesario confirmarlo con el servicio de redes de nuestra organización para evitar problemas de saturación de la red.
- Puerto multicast.
Es el puerto por el que el equipo envía por multicast sincroniza con los clientes. Los valores pueden estar comprendidos entre 9000 y 9050.
- Velocidad.
Es la velocidad máxima medida en Mbits/segundo. El valor máximo admitido por el momento por el motor de clonación es de 999 Mbits/seg.

Modo Multicast	Full-Duplex
IP Multicast	239.255.207.1
Puerto Multicast	9000
Velocidad Multicast	900

Figura 4.36: Datos del protocolo multicast para un aula

Los datos que se han empleado para nuestras aulas son los que se pueden ver en la figura 4.36. Como en las aulas se dispone de electrónica de red a Gigabit se podría haber puesto en la velocidad Multicast la velocidad máxima de 999 Mbits/seg. Se ha puesto un valor inferior para no dejar todo el ancho de banda de red ocupado en los procesos de restauración de una imagen.

Respecto a los datos del formulario de modificación masivo de propiedades de ordenadores que podemos ver en la figura 4.33, principalmente se utiliza el campo de repositorio ya que de esta manera se asigna a todos los ordenadores de ese aula al repositorio que tengamos. No se va a entrar en detalles del resto de campos de este formulario ya que para nuestra organización no es necesario configurar más datos. Quizá se podría forzar a que todos los ordenadores que estén en un aula tengan la misma versión del ogLive. Indicar de nuevo, que mediante este formulario podemos asignar a un conjunto de ordenadores que estén en un aula una serie de propiedades para que se apliquen en conjunto y evitar así tener que ir configurando equipo por equipo.

4.6.3.4. Proyecto RemotePC

Para poder activar este servicio se ha tenido que acudir a la información del proyecto OpenGnsys que se cita en [37].

Son necesarias tres partes fundamentales para este servicio (figura 4.37):

- *Broker* de acceso remoto, que permite al usuario disponer desde su dispositivo y fuera del horario lectivos de los escritorios utilizados en los laboratorios de prácticas. Esta parte se ha tenido que configurar por parte de la universidad.
- Orquestador de servicios (OpenGnsys), que se encargará de ejecutar operaciones de arranque y parada del sistema operativo instalados en los equipos de las aulas TIC, según las sesiones gestionadas por el *broker* y la disponibilidad del parque de ordenadores.

- Ordenadores instalados en los laboratorios, que serán gestionados por el orquestador de servicios.



Figura 4.37: Arquitectura del proyecto RemotePC

Ha sido necesario reconfigurar *MySQL* para su compatibilidad con el proyecto RemotePC. Si se accede a la consola web de OpenGnsys y revisando las propiedades de cualquier aula sobre la que queramos activar RemotePC se muestra el aviso como en la figura 4.38, es necesario activar el calendario de eventos de la base de datos, modificando como usuario *root* el archivo de configuración de *MySQL* situado en `/etc/mysql/mysql.conf.d/mysql.d/mysql.d.cnf` y agregando al final del mismo la línea:

```
event_scheduler = ON
```

Una vez modificado y reiniciado el servicio de *MySQL*, el mismo formulario de propiedades de aula permitirá marchar la casilla de acceso remoto, tal como se puede observar en la figura 4.39.

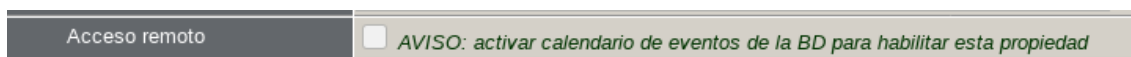


Figura 4.38: Aviso de acceso remoto

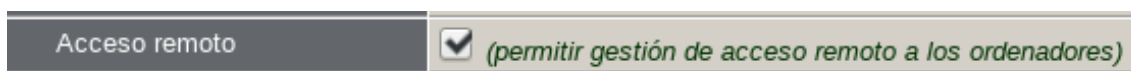


Figura 4.39: Acceso remoto activado

Cosas que se han tenido que hacer para la activación del Proyecto RemotePC:

- Se ha tenido que delegar la administración de la OU AIM a un usuario administrador. La información de dicho usuario (usuario y contraseña) se le ha tenido que dar al administrador del *Broker* de acceso remoto (como se ha comentado antes, esta parte se ha gestionado a través de los servicios del ASIC de la universidad. Además también se le ha tenido que indicar la dirección del servidor OpenGnsys que disponemos.
- Se ha generado un par de grupos de seguridad de una OU que dispongo en el AD de la Universidad (uno para el dominio ALUMNO y otro para el dominio UPV-NET). De esta manera únicamente tendrán acceso a este servicio los alumnos o el personal del departamento que se les quiera conceder acceso a este servicio.

- Se les ha tenido que conceder permisos para poder acceder por escritorio remoto (RDP) a estos grupos que se han creado anteriormente. Para ello he empleado las GPO's de mi OU donde tengo los equipos de las aulas a las que quiero dar acceso (se ha aprovechado que estamos trabajando con máquinas que están en Directorio Activo).

Una vez se tiene configurado todo, el funcionamiento es muy sencillo:

- Introducimos la siguiente URL en un navegador web: <https://polilabs.upv.es>. Deberemos identificarnos como ALUMNO o como UPVNET tal como se muestra en la figura 4.40.
- Una vez identificados (figura 4.41) deberemos instalar el plugin RDP para poder lanzar la aplicación. Una vez instalado el plugin, presionamos sobre la imagen DIMM.
- Se lanza la aplicación UDS, como se puede observar en la figura 4.42.
- Si todo es correcto, lo que va a ocurrir es que desde OpenGnsys se va a arrancar un equipo del aula (este proceso se puede demorar un poco).
- Una vez arrancado el equipo del aula, se establece una conexión a través del protocolo RDP con ese equipo que se ha encendido previamente con las credenciales nuestras (no hace falta volver a introducir las ya que copia las credenciales que se utilizó para entrar en la web de poliLabs).
- Se puede restringir el horario de acceso a través de RemotePC para evitar interferir en el uso de dichos equipos para prácticas docentes que se impartan en las aulas del departamento.

De una manera sencilla se ha permitido a un usuario autorizado acceder a un equipo de trabajo de las aulas de un departamento dedicado a la realización de prácticas docentes fuera de horarios lectivos. Para ello desde OpenGnsys se enciende un equipo aleatoriamente del aula y se establece conexión remota a dicho equipo con las credenciales del usuario.

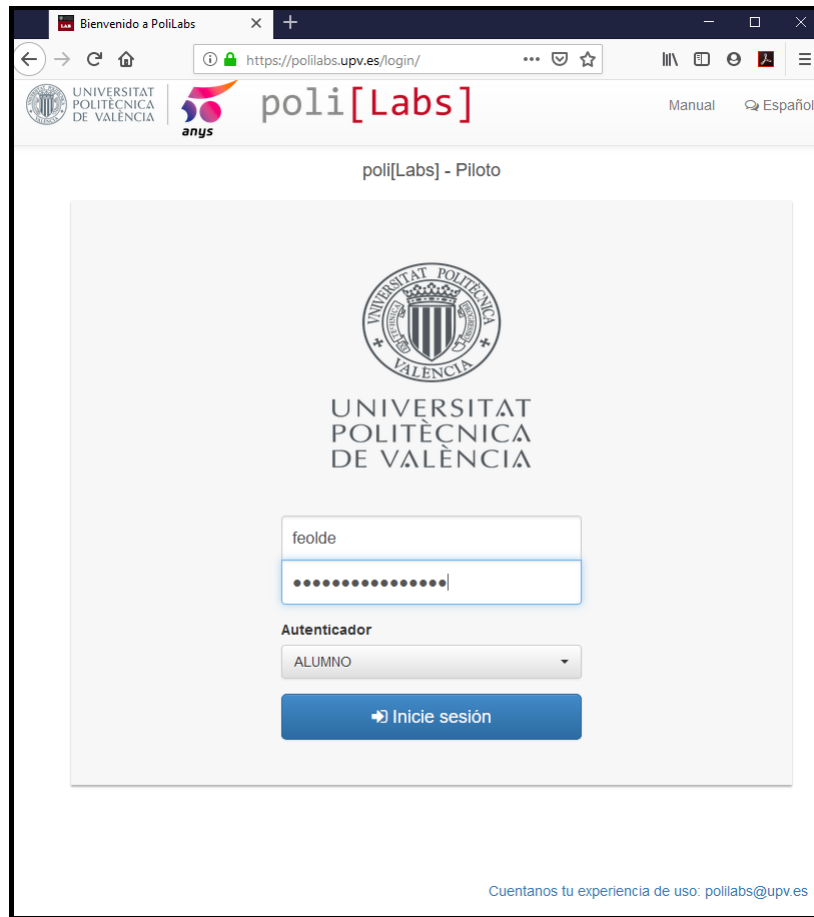


Figura 4.40: Acceso poliLabs

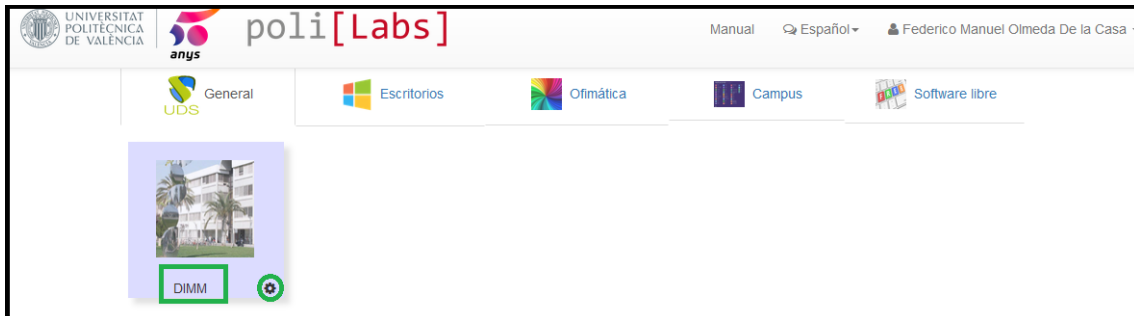


Figura 4.41: Acceso identificado a poliLabs

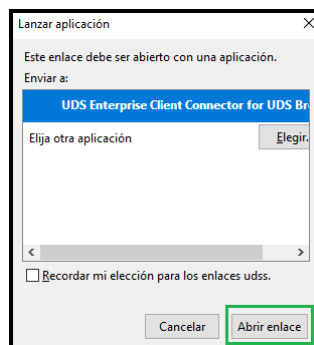


Figura 4.42: Ventana de lanzar la aplicación UDS

4.6.3.5. Gestión de ordenadores

La gestión de ordenadores incluye el alta, modificación de sus propiedades y su eliminación. Se puede mover ordenadores entre distintos ámbitos y realizar una búsqueda para localizar un equipo determinado.

Para dar de alta un equipo se requiere la realización de los siguientes pasos:


- Activar en el ordenador en cuestión el arranque por red.
Para ello se debe entrar en su BIOS y seleccionar el arranque PXE como arranque principal.
- Modificación del servicio DHCP de la red para asignar una IP al equipo.
Es necesario asignar siempre la misma dirección IP al equipo identificado por la dirección MAC de su interfaz de red. En nuestro caso, los equipos están dados de alta en el sistema de la universidad y tienen la IP que les asigna el servidor de DHCP de la universidad.
- Dar de alta el equipo en la aplicación OpenGnsys.

Es necesario también habilitar el WOL o encendido del equipo por red desde la BIOS del mismo así como de cada sistema operativo que tenga el equipo. Disponemos de una variedad de equipos distintos en nuestro departamento y con BIOS diferentes (tenemos equipos servidores, equipos más antiguos, equipos modernos con BIOS más modernos, etc.). El proceso de activación del arranque por red en la BIOS de cada equipo que disponemos no es una tarea trivial por la diversidad de las BIOS. En caso de problemas se ha de consultar el manual del fabricante de la placa base o realizar una búsqueda por internet. Se ha de comentar que por el momento no ha hecho falta esta última opción.

Para dar de alta un equipo en la aplicación OpenGnsys se dispone de dos opciones: Añadir nuevo ordenador o incorporación de ordenadores (en caso de querer añadir varios equipos simultáneamente).

Existe una opción para dar de alta equipos en el servidor DHCP del servidor OpenGnsys por medio de SSH. Como se ha comentado con anterioridad, al estar trabajando con equipos que están dados de alta en la red de la universidad, ya se les asigna una dirección IP única en función de su dirección física MAC. Por tanto no se va a comentar este modo de dar de alta un equipo por medio de DHCP del servidor OpenGnsys pero la información está en [33].

Para dar de alta un nuevo ordenador en nuestra OU AIM se debe seleccionar del menú contextual sobre un aula o un grupo de ordenadores la opción «Añadir nuevo ordenador» y rellenar en el formulario que aparece los parámetros de nombre, IP, MAC y Repositorio manualmente. En la figura 4.43 se puede ver los parámetros al dar de alta un nuevo equipo manualmente desde la consola de la unidad organizativa. Se observa que también se ha seleccionado la versión de ogLive de ese equipo. Se puede modificar un equipo seleccionando la opción «Propiedades» del menú contextual sobre el equipo que se pretenda modificar. Para su eliminación se actual de modo similar pero eligiendo la opción «Eliminar ordenador» del menú contextual.

Nombre *	masteraim-11		(150X110)-(jpg - gif - png) ---- Insertar foto>
Dirección IP	158.42.207.211		
Dirección MAC	E03F49B6A699	Examinar... No se ha seleccionado ningún archivo.	
Nº de serie	(no detectado)		
Foto ordenador	fotoordenador.gif Ver fotos		
Perfil hardware			
Repositorio	AIM_REPO		
ogLive	ogLive-xenial-4.8.0-amd64-r5331		
Menú			
Autoexec			
Interfaz de red	eth0		
Driver de red	generic		
Validación	No		
Página login			
Página validación			

* AVISO: El nombre de equipo no debe superar los 15 caracteres si se instalan sistemas Windows.

Figura 4.43: Datos alta de un equipo nuevo

Para dar de alta un conjunto de ordenadores se ha de seleccionar del menú contextual sobre un aula la opción «Incorporar ordenadores». Se ha de introducir los datos de los equipos en formato DHCP group tal como se puede ver en la figura 4.44 y el resto de los parámetros los hereda del aula.

Incorporación de Ordenadores

 **Aula:Aula2**

Código DHCP
(declaraciones "host")

```
host equipo1 { hardware ethernet 00:27:0E:12:51:E7; fixed-address 158.42.207.1; }
host equipo2 { hardware ethernet 00:27:0E:12:51:E8; fixed-address 158.42.207.2; }
host equipo3 { hardware ethernet 00:27:0E:12:51:E9; fixed-address 158.42.207.3; }
host equipo4 { hardware ethernet 00:27:0E:12:51:E4; fixed-address 158.42.207.4; }
host equipo5 { hardware ethernet 00:27:0E:12:51:E2; fixed-address 158.42.207.5; }
```

AVISO: El nombre de equipo no debe superar los 15 caracteres si se instalan sistemas Windows.
 ATENCIÓN: El equipo iniciará en local por defecto; usar NetBoot Avanzado para modificar su arranque.

✖ Cancel
✔ Accept

Figura 4.44: Incorporación de ordenadores a un aula

4.6.3.6. Gestión de arranque remoto

El servidor OpenGnsys ofrece el servicio PXE, que permite la gestión del arranque remoto de un ordenador cliente tal como se cita en [33].

Cuando un equipo gestionado por OpenGnsys inicia el arranque por red (PXE), conectará con el servicio PXE para consultar e iniciar el tipo de arranque definido para él.

Se dispone de los siguientes arranques definidos en el sistema OpenGnsys:

■ **Arranque PXE en modo usuario «ogLive».**

El arranque nativo de OpenGnsys es un sistema operativo de red basado en Ubuntu que incluye las herramientas necesarias para realizar tareas de particionado, inventariado y clonación.

■ **Arranque PXE en modo administración «ogLiveAdmin».**

Este sistema de ogLive además de realizar las operaciones de inicio en modo usuario realiza también:

- *udpateBootCache*: procedimiento que independiza el tiempo de carga del ogLive del número de ordenadores en arranque simultáneo.
- Inicio en modo depuración ofreciendo una ventana de comandos para ello.
- Configuración del acceso de escritura con el repositorio (en el modo normal o de usuario el montaje es de solo lectura).
- Ofrece una consola de administración.

■ **Arranque en local**

- (mbr 1hd).
El ordenador siempre arrancará según el gestor de arranque definida en el MBR del primer disco duro (hd).
- (1hd 1particion).
El ordenador siempre arrancará desde dicha partición si existe un gestor de arranque instalado en el primer disco duro.
- (1hd 2particion).
El ordenador siempre arrancará desde dicha partición si existe un gestor de arranque instalado en el primer disco duro.

Cuando un equipo es dado de alta en OpenGnsys, su gestor de arranque remoto o «NetBoot» lo incluye automáticamente en un estado de arranque «Sin designar», que es la opción por defecto para todos los nuevos equipos. OpenGnsys define que el modo de arranque «Sin designar» sea un arranque por el MBR del equipo. Se ha de comentar que si en un aula todos los equipos ya están asociados a una opción de arranque, la opción «Sin designar» no aparece.

Para modificar el modo de arranque de uno o varios equipos se ha de entrar en la pestaña de «Aulas» y pulsando con el botón derecho elegimos el menú contextual «NetBoot avanzado» tal como se puede ver en la figura 4.45. El formulario que aparece en la zona de formularios de la zona de trabajo de la consola de OpenGnsys, consta de un contenedor principal (recuadro verde) que representa al arranque principal (ogLive) y varios contenedores auxiliares (recuadro rojo) con botones [OUT][IN] que presentan a cada una de los diferentes tipos de arranques predefinidos.

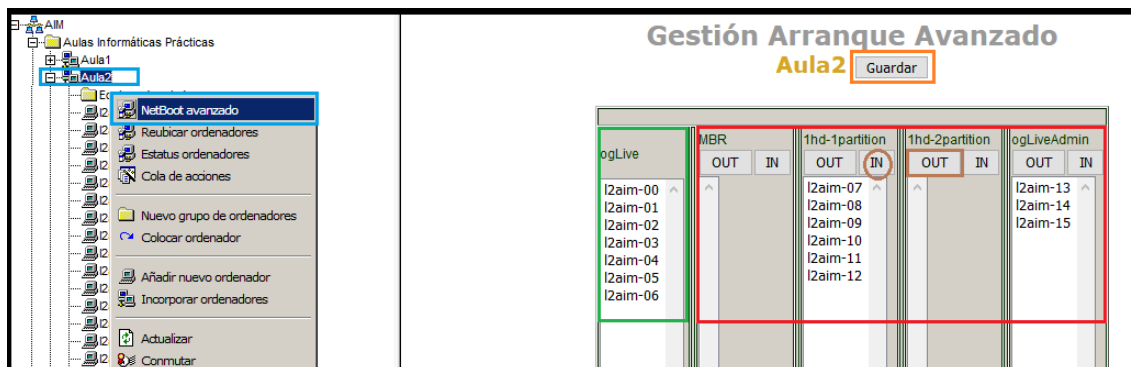


Figura 4.45: Gestión arranque avanzado desde OpenGnsys

El funcionamiento de este formulario es el siguiente:

- La asignación de un tipo de arranque a un ordenador siempre se realiza desde el contenedor principal a un contenedor auxiliar por medio del botón [IN].
- Para asignar un ordenador que está en un contenedor auxiliar al contenedor principal, hay que seleccionarlo y usar el botón [OUT]. En ese momento, el ordenador se habrá colocado en el contenedor principal.
- Si se quiere pasar un equipo entre dos columnas distintas del contenedor auxiliar se ha de presionar el botón [OUT] para sacarlo al contenedor principal, deberemos seleccionar de nuevo el equipo y presionar en el botón [IN] en la columna que nos interesa del contenedor auxiliar.

Una vez se hayan realizado los cambios oportunos para la gestión de arranque avanzado, se debe presionar el botón de «Guardar» para que se apliquen los cambios.

4.6.4. Acceso SSH a los clientes

Desde la consola SSH al servidor OpenGnsys se puede cambiar la clave de acceso Samba y la clave de «root» para poder establecer la conexión SSH a los clientes que se quieran. Para ello se ha de lanzar el siguiente comando:

```
sudo /opt/opengnsys/bin/setsmbpass
```

Con la clave que se ha generado en el comando anterior, ya se puede conectar mediante una consola SSH a los clientes que se quiera acceder. Para ello ese equipo ha de estar arrancado con el sistema operativo PXE ogLive. Una vez conectado por SSH, se tendrá la estructura de directorios de un sistema operativo linux y se podrá acceder al contenido de ese equipo (se puede acceder a su disco duro, borrar y copiar archivos, etc.).

4.7 Estado de los ordenadores

Un ordenador gestionado por OpenGnsys puede estar en alguno de estos estados: iniciado en un sistema operativo Windows, GNU/Linux o Mac Os, esperando o realizando operaciones desde el sistema de arranque PXE ogLive o apagado. Desde la interfaz web se puede mostrar el estado de los ordenadores al seleccionar la opción «Estatus ordenadores» de cualquier ámbito tal como se puede observar en la figura 4.46.

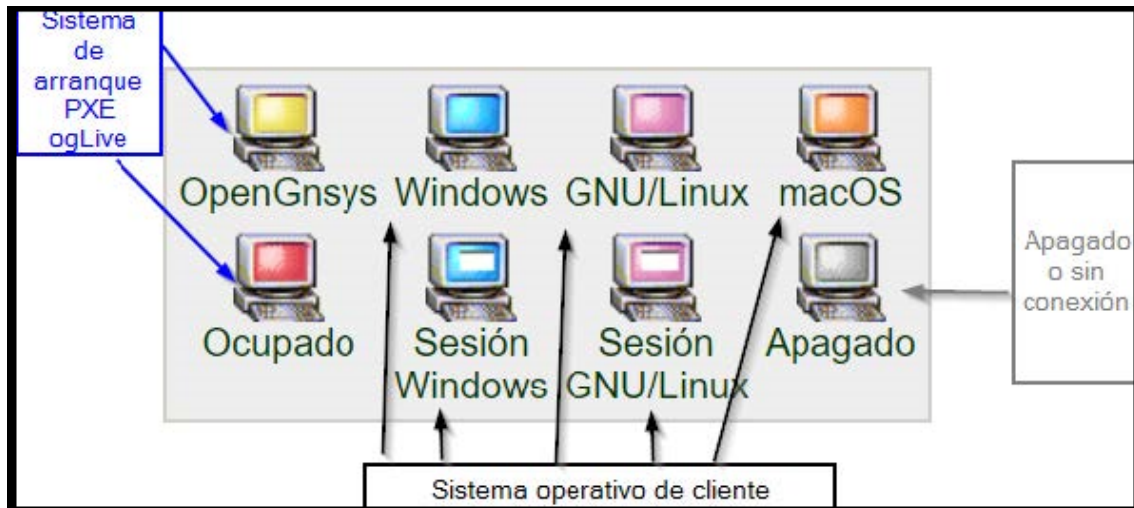


Figura 4.46: Estado de los equipos cliente gestionados por OpenGnsys

OpenGnsys ofrece unos comandos básicos para la gestión de los ordenadores clientes (su disponibilidad depende del ámbito sobre el que se ejecuten). Además, los comandos pueden ser incluidos en la «cola de acciones» para confirmar que la ejecución sea diferida (cuando el cliente esté iniciado en el cliente específico PXE o tras realizar una operación en curso). El resultado de las operaciones o acciones se puede consultar desde la opción «log histórico» o visualizar los detalles a tiempo real de una operación desde la opción «log a tiempo real».

OpenGnsys dispone para cada tipo de sistema operativo de usuario un agente que se comunica con el servidor y que se debe instalar en el equipo modelo. De esta manera se permite que desde la aplicación web de OpenGnsys se puede mostrar el estado de los equipos de un aula y enviar ciertos comandos a los equipos.

4.7.1. Instalación del agente en el sistema operativo de cliente

El ogAgent se descarga desde la propia consola web de OpenGnsys. Para ello introducimos la siguiente URL desde un navegador del equipo cliente que utilizamos como equipo base para la clonación en el resto de equipos: <https://nombreservidorOpenGnsys/opengnsys/descargas/>.

Se procede con la descarga del fichero README donde se indicarán las particularidades de la instalación o los requisitos previos. Seleccionamos el binario que nos interese en función del sistema operativo, tal como se puede ver en la tabla 4.5.

Sistema Operativo	OGAgent	Dependencias	Fichero configuración
MS Windows	.exe	NO	Para sistemas de 64 bits: C:\Program Files (x86)\OGAgent\cfg\ogagent.cfg Para sistemas de 32 bits: C:\Program Files\OGAgent\cfg\ogagent.cfg
macOS X	.pkg	SI	/Applications/OGAgent.app/cfg/ogagent.cfg
Ubuntu, Debian y derivados	.deb	SI	/usr/share/OGAgent/cfg/ogagent.cfg
Red Hat, Fedora y derivados	.rpm	NO	

Tabla 4.5: Información de biarios para los agentes a instalar en los sistemas operativos clientes

Tras haber instalado el ogAgent descargado en el sistema operativo cliente, hay que configurarlo mediante el fichero «ogagent.cfg» (la ruta de donde se encuentra se muestra en la tabla anterior). Para ello hay que editar el fichero y modificar la siguiente línea (en vez de poner «IPServidorOpenGnsys» hay que poner la dirección IP de la misma):

```
# Remote OpenGnsys Service
remote=https://IPServidorOpenGnsys/opengnsys/rest
```

Una vez realizado la configuración en el agente, se observa que desde la web de OpenGnsys visualizamos el estado de los equipos que acabadmos de configurar.

4.7.2. Ámbito de aplicación y modo de ejecución

Es necesario saber sobre qué equipos se va a realizar una determinada acción. El **ámbito de aplicación** determina qué objetos de tipo «ordenador» de la estructura jerárquica de OpenGnsys debe realizar una operación solicitada. Puede ser una unidad organizativa, aula, grupo de aulas, grupo de ordenadores o un ordenador.

El **modo de ejecución** determina cómo serán ejecutadas las operaciones por los ordenadores. Existen dos modos:

- Directo (es inmediato).
Se realizará siempre y cuando el servidor de OpenGnsys pueda comunicarse con el correspondiente agente del sistema operativo de usuario y la operación esté habilitada para ser realizada por ese agente.

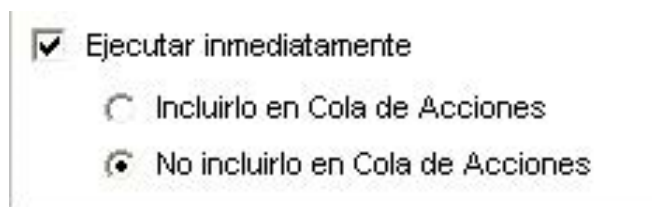


Figura 4.47: Modo de ejecución directo

- Diferido (gestionado por la cola de acciones).
Se realizará cuando el agente capaz de procesar la operación esté activo.

Ejecutar inmediatamente
 Incluirlo en Cola de Acciones
 No incluirlo en Cola de Acciones

Figura 4.48: Modo ejecución diferido

- Modo directo para los agentes conectados y diferido para los agentes apagados.

Ejecutar inmediatamente
 Incluirlo en Cola de Acciones
 No incluirlo en Cola de Acciones

Figura 4.49: Modo ejecución directo y diferido

- Sin ejecución (no realiza la petición).

Ejecutar inmediatamente
 Incluirlo en Cola de Acciones
 No incluirlo en Cola de Acciones

Figura 4.50: Modo sin ejecución

4.7.3. Operaciones o acciones básicas: los comandos

Desde la consola web de OpenGnsys se puede solicitar operaciones o acciones sobre los ordenadores que se gestionan. Estas operaciones puede ser «comandos» y «asistentes» (son acciones que requieren parámetros de ejecución que los proporciona un usuario administrador). En las operaciones simples, ya sean comandos o asistentes, se pueden agrupar como:

- Procedimientos.
 - El ámbito de aplicación es seleccionado por el usuario en el momento de la solicitud.
 - No se puede filtrar los equipos de un ámbito de aplicación.
 - El modo de ejecución siempre es diferida (gestionada por la cola de acciones). Se trata pues de una ejecución directa e incluida en la cola de acciones.
- Tareas.
 - El ámbito de aplicación está vinculado a la tarea como una propiedad más.
 - No se puede filtrar los equipos de un ámbito de aplicación.
 - El modo de ejecución siempre es diferida (gestionada por la cola de acciones). Se trata pues de una ejecución directa e incluida en la cola de acciones.

El formulario de ejecución de un comando permite la selección de los objetos de tipo ordenador sobre los que se va a ejecutar la operación. Esta selección solo es efectiva sobre una ejecución directa (nunca sobre una ejecución diferida). Dicha selección puede ser manual, eligiendo los equipos uno a uno, o de manera automática utilizando los filtros como se puede ver en la figura 4.51.

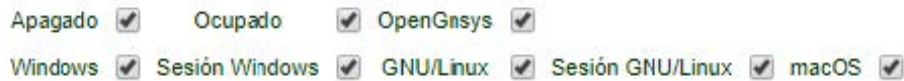


Figura 4.51: Filtro de ejecución de un comando basado en su estado

Los comandos predefinidos específicos son:

- Para todos los agentes: apagar, arrancar, reiniciar, ejecutar script.
- Para los agentes de los sistemas operativos de cliente: enviar mensaje.
- Para el agente ogLive: crear imagen, iniciar sesión, inventario hardware, inventario software, restaurar imagen y particionar y formatear.

4.7.4. Supervisión y resultado de operaciones

Se dispone de varios mecanismos para ver el resultado de las operaciones realizadas sobre un ámbito de ejecución:

- Cola de acciones.
Se lanza desde el menú contextual sobre un ámbito de ejecución de un aula «Cola de acciones», tal como vemos en la figura 4.52.
- Log en tiempo real.
Se lanza desde el menú contextual sobre un equipo de un aula «Log en tiempo real».
- Log histórico.
Se lanza desde el menú contextual sobre un equipo de un aula «Log histórico».

Cola de acciones

Opciones de búsqueda

Acciones	Resultados
Todas <input type="button" value="v"/>	Todos <input type="button" value="v"/>
Estados <input type="button" value="v"/>	Porcentaje desde: 0 hasta: 100
Todos <input type="button" value="v"/>	
Fecha inicio: 03/02/2019	Fecha final: 03/08/2019
Hora inicio: <input type="text"/>	Hora final: <input type="text"/>

Opciones de visualización

Ver parámetros Ver subprocesos Ver Notificaciones

🔍

Ámbito: Aulas, Aula2 🖨

Eliminar
 Reiniciar
 Detener
 Reanudar
 Finalizar sin errores
 Finalizar con errores

	R	Fecha final	Hora final	Fecha inicio	Hora inicio	Ámbito	Información	S	%
	✓	15-07-2019	23:17:18	15-07-2019	21:09:35	Aulas			100%
Restaurar Imagen	✓	15-07-2019	23:17:18	15-07-2019	21:09:35	I2aim-12			
Restaurar Imagen	✓	15-07-2019	23:17:18	15-07-2019	21:09:35	I2aim-09			
Restaurar Imagen	✓	15-07-2019	23:17:16	15-07-2019	21:09:35	I2aim-08			
Restaurar Imagen	✓	15-07-2019	23:17:17	15-07-2019	21:09:35	I2aim-07			
Restaurar Imagen	✓	15-07-2019	23:17:16	15-07-2019	21:09:35	I2aim-06			
Restaurar Imagen	✓	15-07-2019	23:17:17	15-07-2019	21:09:35	I2aim-05			
Restaurar Imagen	✓	15-07-2019	23:17:16	15-07-2019	21:09:35	I2aim-04			
Restaurar Imagen	✓	15-07-2019	23:17:18	15-07-2019	21:09:35	I2aim-03			
Restaurar Imagen	✓	15-07-2019	23:17:17	15-07-2019	21:09:35	I2aim-01			
Restaurar Imagen	✓	15-07-2019	23:17:16	15-07-2019	21:09:35	I2aim-00			
Restaurar Imagen	✓	15-07-2019	23:17:17	15-07-2019	21:09:35	I2aim-02			
	✓	15-07-2019	21:08:34	15-07-2019	21:08:02	Ordenadores			100%
Ejecutar Script	✓	15-07-2019	21:08:34	15-07-2019	21:08:02	I2aim-06			
	✓	15-07-2019	21:08:26	15-07-2019	21:07:53	Ordenadores			100%
Ejecutar Script	✓	15-07-2019	21:08:26	15-07-2019	21:07:53	I2aim-05			

Figura 4.52: Cola de acciones sobre el ámbito Aula 2

4.8 Particionado básico

El proceso de particionado de los discos duros es el más delicado de todos los que se pueden realizar desde OpenGnsys, ya que cualquier error puede ocasionar que se quede el ordenador cliente inoperativo debido a que se eliminen los sistemas operativos existentes o que el proceso de restauración no finalice correctamente por haber definido un tamaño o tipo de disco inadecuado.

La partición cache es una partición especial de OpenGnsys que tiene varias funciones:

- Proporcionar un contenedor intermedio de imágenes.
Se trata de un contenedor intermedio entre el repositorio y la partición destino del sistema operativo. Además su uso permite disminuir los tiempos de restauración y es el soporte auxiliar para las transferencias P2P.
- Servir de contenedor intermedio para el sistema operativo PXE ogLive.
La partición cache puede alojar la primera fase del ogLive con lo que se independiza el tiempo de inicio del ogLive de la carga de red o del servidor OpenGnsys. En los casos que se inicie el sistema operativo PXE ogLiveAdmin se realiza de manera automática la activación del ogLive en la cache.

Se trata de una partición Linux con sistema de archivos EXT4 que se ha de ubicar siempre en una partición primaria ubicada en la cuarta posición de cualquier disco duro. El tamaño para esta partición depende del uso que se le vaya a dar (al menos debe tener un tamaño de 1 GB de capacidad). Si se pretende almacenar la imagen del sistema para que la restauración sea mucho más rápida, se debe incrementar notablemente dicho espacio.

OpenGnsys ofrece un dos posibilidades para la realización de tareas de particionado [38]:

- Comando «Particionar y formatear». Trabaja a nivel de particiones primarias del primer disco duro MSDOS. Se permite crear o eliminar una partición concreta y formatear el sistema de archivos de manera individual.
- Asistente de particionado. Esta opción es más compleja. Permite trabajar con varios discos, crear tablas de particiones de discos MS-DOS [39] y GPT [40], crear particiones primarias y extendidas, no formatea las particiones, etc..

A pesar de disponer de las herramientas comentadas anteriormente, a la hora de particionar los discos se ha preferido realizar mediante *scripts* el modo de creación de las particiones de los discos. Para el caso concreto de las aulas informáticas se dispone de un disco SSD de 512GB de capacidad y un disco SATA de tamaño variado en función del aula en la que nos encontremos.

El script que se ha utilizado ha sido el siguiente:

```
ogEcho session "Se procede al particionado de los discos del aula 1 –320 GB y
512 SSD"
ogEcho session "Se procede a crear la particion de 485GB en el disco de 512GB
SSD –disco 1–"
ogUnmountAll 1 >/dev/null
ogCreatePartitionTable 1 MSDOS
ogDeletePartitionTable 1
ogUpdatePartitionTable 1
ogEcho session "Se ha borrado el disco 1"
ogEcho session "Creamos particion WINDOWS en disco 1"
ogExecAndLog command session ogCreatePartitions 1 NTFS:485000000
ogSetPartitionActive 1 1
ogUpdatePartitionTable 1
ogEcho session "Se procede a crear la cache de 250GB en el disco de 500GB –
disco 2–"
ogUnmountAll 2 >/dev/null
ogUnmountCache
ogCreatePartitionTable 2 MSDOS
ogDeletePartitionTable 2
ogUpdatePartitionTable 2
ogEcho session "Se ha borrado el disco 2"
ogEcho session "Creamos la particion cache"
ogCreateCache 2 4 250000000
ogFormatCache
ogUpdatePartitionTable 2
ogEcho session "Fin creacion cache"
ogEcho session "Fin creacion particion WINDOWS en disco 2"
```

Lo que se ha hecho es seleccionar un equipo de un aula y lanzar el comando «Ejecutar script». Se ha puesto el código en la ventana de código del script y se ha guardado como

un procedimiento indicándole un nombre adecuado para poder reutilizarlo (se ha llamado «ParticionarDiscosAula2»). Una vez lanzado el código se debe haber particionado correctamente los discos de ese equipo sobre el que se ha lanzado el comando. Teniendo ya verificado el correcto funcionamiento del procedimiento, se puede aplicar al ámbito que se desee. Cabe mencionar el hecho de que si se restaura una imagen sobre una partición sin formatear, el proceso finaliza correctamente, ya que el sistema de archivos está incluido dentro de la imagen y se restaura en la partición.

4.9 Crear una imagen del sistema

OpenGnsys realiza la clonación de equipos mediante imágenes del sistema de ficheros. La imágenes puede ser monolíticas (guardan el contenido del sistema de ficheros en un archivo y solo permiten crearlo o restaurarlo en su totalidad) o sincronizadas (imágenes diferenciales creadas a partir de una imagen básica). Esta información se ha obtenido de [41].

Lo primero que se ha de hacer es preparar el cliente modelo. Se recomienda una serie de pasos para la instalación de los sistemas operativos:

- Particionar el equipo de referencia con OpenGnsys.
De esta manera se evita que el sistema operativo utilice más de una partición (en Windows).
- Poner todo el sistema operativo en una sola partición.
- No usar LVM, RAID o discos dinámicos en Windows.
- Gestor de arranque Grub en la partición root y no en el MBR.

Con respecto a los controladores hardware, en la medida de lo posible, en los sistemas operativos Windows se deben instalar los drivers para los dispositivos que tengan los distintos equipos que vayan a compartir la misma imagen para evitar tener que crear distintas imágenes por el hardware de los equipos.

Se ha de personalizar el sistema operativo:

- Poner un nombre de equipo cuya longitud máxima sea de 15 caracteres.
Cuando se restaura la imagen en un equipo, se cambia el nombre en el sistema operativo asignándole el mismo nombre que tenga el equipo en OpenGnsys.
- Los parámetros de red asignados por DHCP.
- Limpieza del sistema operativo.
Se recomienda eliminar los datos que no son necesarios: ficheros temporales, historial-cache-descargas-cookies de los navegadores web, datos temporales de los usuarios, etc. En el proceso de creación de imágenes de OpenGnsys se borran los ficheros de paginación (*pagefile.sys*) y de hibernación (*hiberfil.sys*).
- Comprobación del sistema de archivos.
Se recomienda en Windows ejecutar las herramientas de liberación de espacio, desfragmentación y comprobación de disco.

Se ha de instalar el agente de OpenGnsys para los sistemas operativos cliente. Además si estamos con un sistema operativo Windows se debe desde una consola de símbolo de sistema como administrador realizar algunas acciones:

- Generalizar el gestor de arranque de Windows.

```
bcdedit /set {current} osdevice boot
bcdedit /set {current} device boot
bcdedit /set {bootmgr} device boot
bcdedit /set {memdiag} device boot
```

- Desactivar el inicio rápido.

```
reg ADD "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Contro\Session
Manager\Memory Management\PrefetchParameters" /v EnableSuperfetch /t
REG_DWORD /d 0x3 /f
```

4.9.1. Inventario de hardware del equipo modelo

Desde OpenGnsys se puede gestionar la información del hardware y del software de los distintos equipos. Los datos de los dispositivos de hardware de un determinado equipo se almacenan en el «Perfil de Hardware» (como se puede observar en la figura 4.53) mientras que los sistemas operativos y sus aplicaciones se guardan en el «Perfil de Software». El hecho de catalogar el inventario de hardware de los equipos controlados por OpenGnsys es una buena prácticas de uso que ayuda al administrador de OU a conocer los drivers de los dispositivos que posee un equipo que se ha de instalar antes de la creación de la imagen del sistema. El comando para realizar el inventario hardware de un equipo es «Inventario Hardware». Para lanzarlo sobre un equipo, se debe seleccionar el equipo desde la consola Web de OpenGnsys y con el menú contextual de ese equipo, seleccionar el comando indicado anteriormente. Se puede consultar la información del hardware y software de los equipos desde los botones de la barra de herramientas «Hardware» y «Software» respectivamente.

The screenshot displays the OpenGnsys web interface. On the left, a navigation tree shows 'Hardware' > 'Perfiles hardware' > 'Perfil hardware (11aim-15)'. The main content area is titled 'Perfiles hardware' and 'Information sobre Perfiles hardware'. It shows a detailed list of hardware components for the selected profile:

Información Perfil hardware	
Perfil hardware (11aim-15)	Información Perfil hardware
Componentes hardware	
(Placas) ASUSTeK COMPUTER INC. H81M-PLUS v.Rev X.0x	
(Dispositivos Multimedia) Intel Corporation 8 Series/C220 Series Chipset High Definition Audio Controller v.05	
(Dispositivos Multimedia) Intel Corporation Xeon E3-1200 v3/4th Gen Core Processor HD Audio Controller v.06	
(Tarjetas de Red) Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller 1Gb/1s v.0c	
(Microprocesadores) Intel Corp. Intel(R) Core(TM) i3-4150 CPU @ 3.50GHz 3499MHz v.intel(R) Core(TM) i3-4150 CPU @ 3.50GHz	
(Memorias) Conexant (Rockwell) BLS4G3D1609DS1S00. 4GB 1600MHz (ChannelA-DIMM0)	
(Memorias) Conexant (Rockwell) BLS4G3D1609DS1S00. 4GB 1600MHz (ChannelB-DIMM0)	
(Tarjetas gráficas) Intel Corporation 4th Generation Core Processor Family Integrated Graphics Controller v.06	
(Discos) INTEL SSDSC2KW51 476GiB v.004C	
(Discos) Toshiba TOSHIBA DT01ACA0 465GiB v.A750	
(Marca y modelo del equipo) ASUS All Series (All) v.System Version	
(Controladores USB) ASMedia Technology Inc. ASM1042A USB 3.0 Host Controller v.00	
(Controladores USB) Intel Corporation 8 Series/C220 Series Chipset Family USB EHCI #1 v.05	
(Controladores USB) Intel Corporation 8 Series/C220 Series Chipset Family USB EHCI #2 v.05	
(Controladores USB) Intel Corporation 8 Series/C220 Series Chipset Family USB xHCI v.05	
(Bus del Sistema) Intel Corporation 8 Series/C220 Series Chipset Family SMBus Controller v.05	
(Chasis del Sistema) Desktop	
(Controladores de almacenamiento) Intel Corporation 8 Series/C220 Series Chipset Family 6-port SATA Controller 1 v.05	
(Tipo de proceso de arranque) BIOS	
Ordenadas con este perfil	

Figura 4.53: Información del perfil hardware de un equipo

Una vez se ha creado el perfil de hardware de un equipo modelo, se puede asociar dicho perfil al resto de clientes que tengan la misma distribución y tipo de dispositivos. Desde la consola de administración web se puede asociar un perfil de hardware a un equipo determinado o asignarlo a todos los equipos de una misma aula.

4.9.2. Inventario de software del equipo modelo

Un perfil de software es el catálogo completo y detallado de las aplicaciones y programas instalados en un determinado sistema operativo. Se debe generar un perfil software para cada imagen que se vaya a crear (con el comando «Crear Imagen» se actualiza el perfil de software del sistema operativo a clonar).

La información sobre los perfiles de software se gestiona desde la pestaña «Software» de la barra de menús de la consola web de OpenGnsys. En ella se dispone de tres tipos de objetos:

- **Tipos de software.** Se clasifica el software instalado en aplicaciones, archivos y sistemas operativos.
- **Componentes de software.** Se muestran datos técnicos de todos los tipos de software que se han detectado.
- **Perfiles de software.** Se listan los perfiles de software creados.

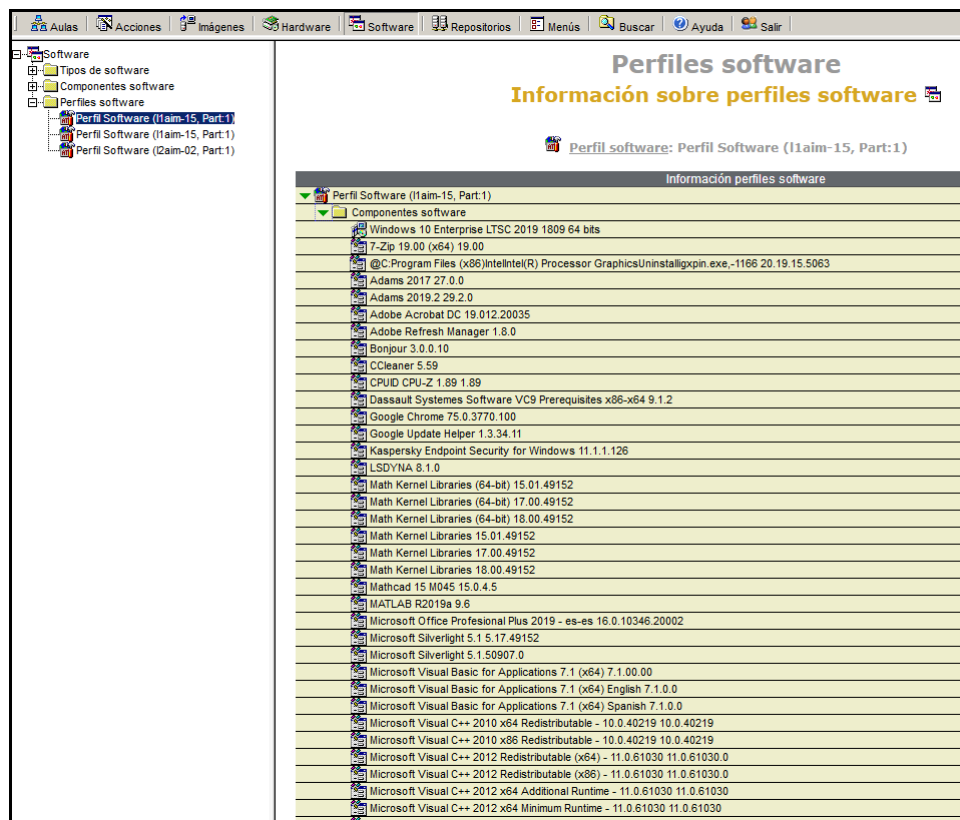


Figura 4.54: Información del perfil software de un equipo

Como se puede observar en la figura 4.54, se muestra la información de Perfil software (11aim-15, Part:1) con una lista de componentes software instalados (aplicaciones Windows).

4.9.3. El objeto imagen monolítica

Como paso previo a la copias de los datos al repositorio en el proceso de creación del fichero imagen, se debe definir un objeto de tipo Imagen en la consola web (de tipo monolítica) que incluirá los datos a utilizar en el proceso de creación física de la imagen.

El administrador de la OU puede acceder a la lista de Imágenes Monolíticas en la consola web de OpenGnsys desde la pestaña del la barra de herramientas «Imágenes», tal como se puede observar en la figura 4.55.

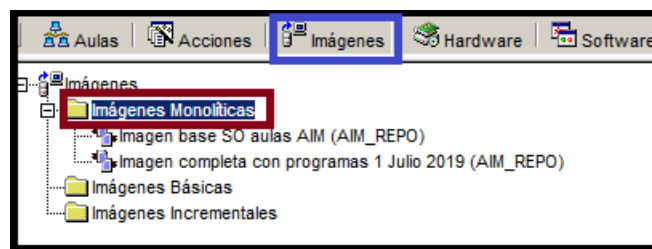


Figura 4.55: Pestaña Imágenes de la consola web OpenGnsys

Un objeto Imagen de tipo monolítica dispone de los siguientes datos tal como se puede observar en la figura 4.56:

- **Nombre.** Es el nombre del fichero de imagen que se va a almacenar en el repositorio o en la cache local del cliente (solo se permiten caracteres alfanuméricos).
- **Descripción.** Nombre completo de la imagen que se va a mostrar en la consola web y que nos permite identificar la imagen.
- **Repositorio.** Donde se va a almacenar la imagen.
- **Comentarios.** Datos descriptivos sobre la imagen.
- **Acceso remoto.** Permite que esta imagen sea accesible mediante RemotePC.
- **Equipo modelo.** Se rellena automáticamente tras la creación de la imagen.
- **Disco, partición.** Se rellena automáticamente tras la creación de la imagen.
- **Fecha de creación.** Se rellena automáticamente tras la creación de la imagen.
- **Perfil software.** Perfil asociado con los datos del sistema operativo instalado.
- **Sistema operativo.**

Gestión Imágenes Monolíticas

Modificar

Nombre	ImagenFullAIM2019
Descripción	Imagen completa con programas 1 Julio 2019
Repositorio	AIM_REPO
Comentarios	El software requerido inicialmente para el curso 2019-2020
Acceso remoto	<input checked="" type="checkbox"/> (permitir gestión de acceso remoto a los ordenadores)
Equipo modelo	I1aim-15 (Aula1)
Disco, Partición	2, 1 (0-EMPTY)
Fecha de creación	2019-07-05 20:45:54 (r3)
Perfil Software	Perfil Software (I1aim-15, Part:1)
Sistema operativo	Windows 10 Enterprise LTSC 2019 1809 64 bits
Datos del repositorio	
Camino	/ImagenFullAIM2019.img
Tamaño	71.96 GB
Copia de seguridad	78.81 GB

ATENCIÓN.- Una imagen que tiene perfil software indica que en algún momento se ha creado y por tanto no pueden modificarse ciertos datos a menos que la elimine y la vuelva a crear




Figura 4.56: Propiedades de un objeto de tipo imagen monolítica

4.9.4. El comando «Crear imagen» de la consola web

El proceso de creación de imágenes tiene cierta complejidad y está formado por varias tareas (tanto en el cliente como en el servidor). Las tareas que se realizan en el cliente son: actualizar el inventario de software del sistema operativo, reducir el sistema de archivos al mínimo ocupado por los datos, crear una imagen comprimida en el repositorio y restablecer el tamaño original del sistema de archivos. Se trata de un proceso largo y mientras se está ejecutando el ordenador modelo no puede ser utilizado. En el equipo se muestra información sobre la ejecución del comando y los tiempos estimados de las distintas tareas.

Las tareas a realizar en el servidor son: bloquear la imagen para uso exclusivo, genera los ficheros de integridad del fichero, genera los ficheros asociados al protocolo P2P y liberar la imagen para que pueda ser usada. Es un proceso largo y durante su ejecución no se puede utilizar la imagen.

4.9.5. Formulario de creación de imagen

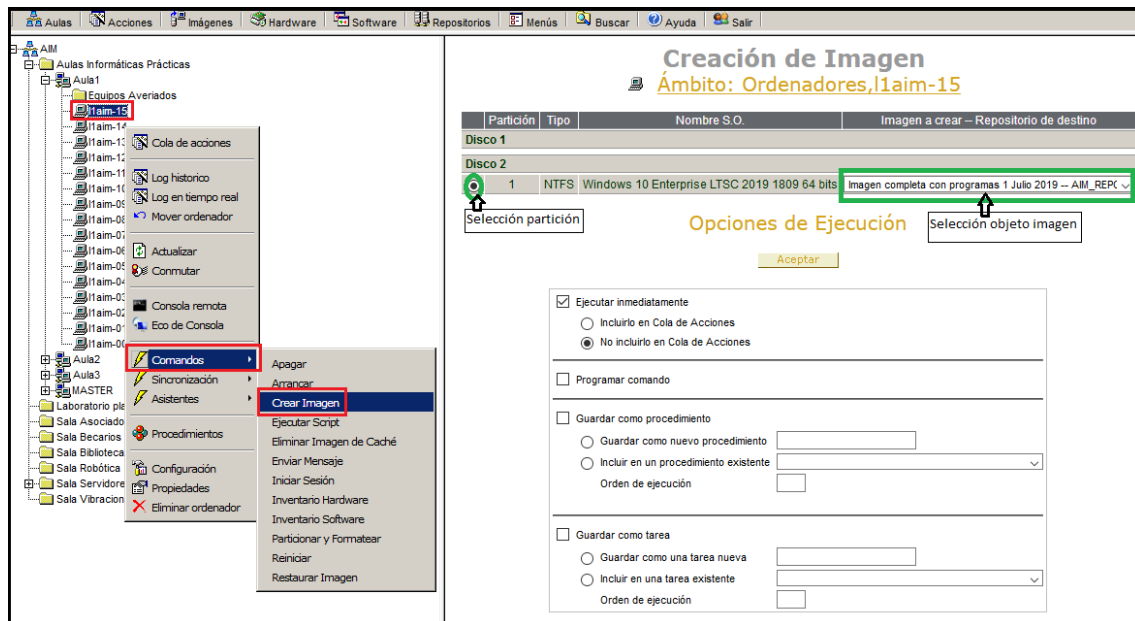


Figura 4.57: Formulario de creación de imagen con OpenGnsys

En la figura 4.57 se muestra el acceso al formulario de creación de una imagen desde OpenGnsys. Para ello se ha tenido que ir a la pestaña «Aulas» de la barra de herramientas, elegir un equipo de un aula que va a servir como equipo modelo para la creación de la imagen del sistema. Con el menú contextual sobre el equipo seleccionado se debe emplear el comando «Crear Imagen». En ese momento debe aparecer el formulario de creación de imagen correspondiente en el que deberemos seleccionar la partición que se desea clonar y elegir el objeto imagen donde se va a realizar la imagen (previamente se habrá tenido que crear dicha imagen tal como se ha visto en la sección 4.9.3. Cuando presionamos sobre el botón de aceptar para que se ejecute la acción se pondrá a crear la imagen del sistema de archivos que se ha elegido. Conviene recordar que este proceso siempre se hace en unicast (la comunicación entre el equipo y el servidor OpenGnsys).

Una vez iniciado el proceso de creación de imagen, desde la consola web de OpenGnsys se puede ver dos tipos de registros de incidencias que el administrador de la OU puede revisar:

- **Log Histórico:** registro completo de incidencias del equipo que se pueden utilizar para comprobar tanto las operaciones en ejecución como las tareas que han concluido.
- **Log en tiempo real:** registro de cambios en tiempo real para la operación que se está ejecutando en el equipo.

El acceso a los logs se realiza desde el menú contextual del objeto ordenador (se puede hacer desde el estado de los equipos de un aula o desde el árbol de objetos).

4.10 Despliegue de una imagen del sistema

En la sección 4.9.3 se mostró cómo crear una imagen del sistema de un equipo modelo desde OpenGnsys. Esta imagen se puede almacenar en un fichero imagen ubicado en el repositorio de OpenGnsys o en local (en la partición cache). Cuando se realiza un clonado desde una imagen creada a una partición de un equipo destino se dice que se ha realizado una **restauración de imagen**. Sin embargo, cuando se quiere hacer sobre las particiones de una serie de equipos, entonces se dice que se ha realizado un **despliegue de imágenes** [42].

El despliegue de imágenes implica dos procesos:

1. Ubicar el fichero imagen lo más cerca del equipo destino (normalmente en el repositorio local o cache del equipo). Este proceso se denomina *updateCache*.
2. Restaurar la imagen.
Si ya tenemos la imagen en la partición cache del equipo destino, entonces se realiza la restauración de la imagen desde el contenedor local cache a la partición destino.

Para la transmisión de datos desde el origen a uno o varios equipos destinos, se emplean varios protocolos de transferencia:

- **unicast.**

Se emplea este protocolo de transferencia en los procesos de creación de la imagen desde el equipo modelo al repositorio del sistema. También se puede utilizar en los procesos de restaurar desde el repositorio del sistema o al realizar un despliegue. En la figura 4.58 se muestra este concepto de transferencia de datos.

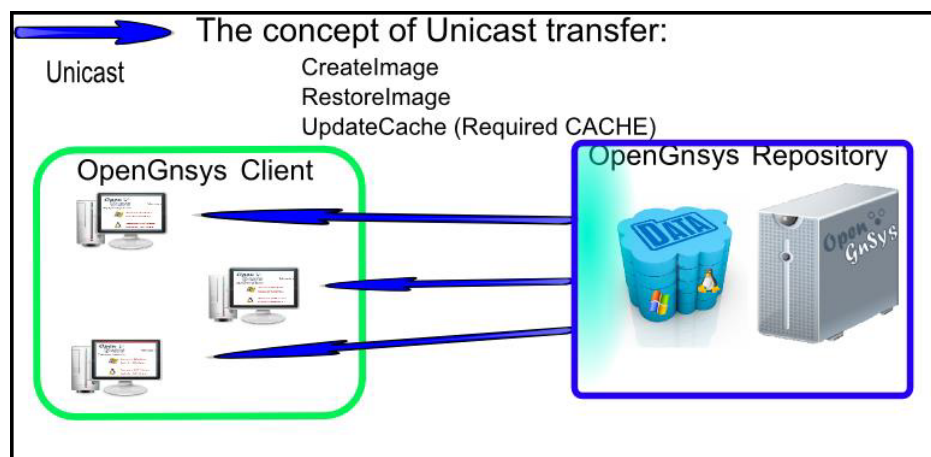


Figura 4.58: Concepto de transferencia de datos unicast

- **multicast.**

Se utiliza en los procesos de restauración de una imagen desde el repositorio central y en los procesos de despliegue. La ventaja principal está en que el repositorio envía una sola vez los datos y son recibidos simultáneamente por los clientes. Además se puede emplear multicast entre subredes diferentes (el repositorio está en una subred y los equipos clientes en otra). Es necesario antes de utilizar este protocolo de comunicación hablar con el administrador de red para verificar su correcta configuración. En la figura 4.59 se muestra este concepto de transferencia de datos.

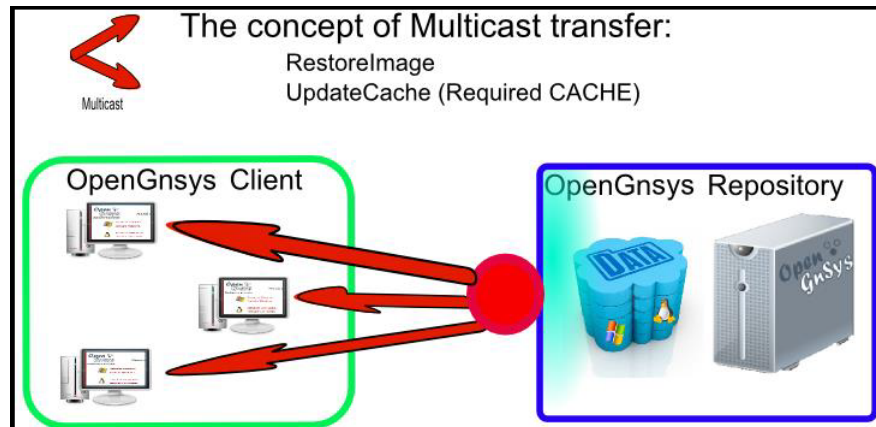


Figura 4.59: Concepto de transferencia de datos multicast

- **P2P.**

Se puede emplear este protocolo siempre y cuando además de disponer de un repositorio central, los clientes disponga del repositorio local o cache. Solo se puede usar en los despliegues de imágenes. En la figura 4.60 se muestra este concepto de transferencia de datos.

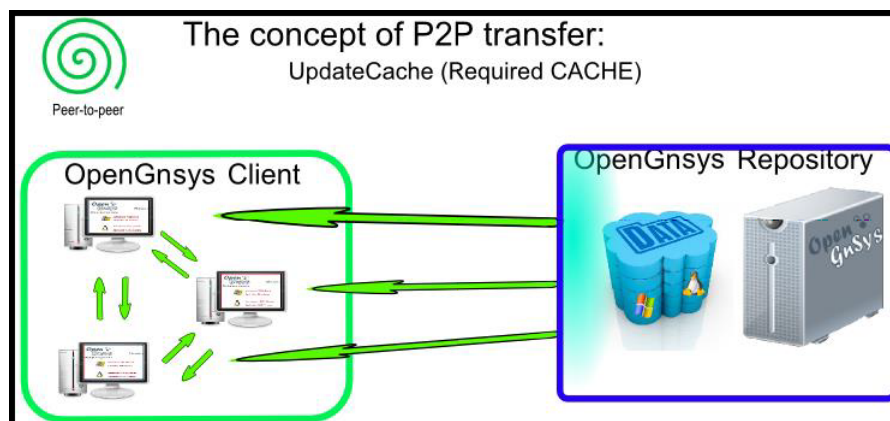


Figura 4.60: Concepto de transferencia de datos P2P

Tanto la clonación como la gestión de imágenes son tratadas a nivel de sistema de archivos. Hay que pensar que la partición del equipo destino ha de ser al menos un poco superior al tamaño de los archivos de referencia.

4.10.1. Comando Restaurar

Desde la consola web de OpenGnsys se puede hacer de una manera sencilla las operaciones de creación y restauración de imágenes de tipo monolíticas. Mediante el comando «Restaurar» se implementa de manera transparente los procesos de restauración, actualización de la cache y el despliegue de una imagen. Existen varios métodos de restauración que se ha de realizar:

- **unicast-cache.**

El despliegue de la imagen usa el repositorio local cache. La imagen se descarga (mediante unicast) al repositorio local cache del cliente y luego se restaura la partición desde esa cache.

- **unicast-direct.**
El despliegue es directo sin utilizar el repositorio local. La imagen se restaura directamente desde el repositorio.
- **multicast-cache.**
El despliegue de la imagen utiliza el repositorio local cache. La imagen se descarga (multicast) al repositorio local cache del cliente y luego se restaura la partición desde esa cache.
- **multicast-direct.**
El despliegue es directo sin utilizar el repositorio local. La imagen se restaura directamente desde el repositorio.
- **torrent-cache.**
El despliegue de la imagen usa el repositorio local cache. La imagen se descarga (Torrent o P2P) al repositorio local cache del cliente y luego se restaura la partición desde esa cache.

En la figura 4.61 se observa el comando «Restaurar» aplicado sobre un equipo. En el formulario que aparece primero se ha de seleccionar la partición del equipo destino donde se quiere realizar la restauración, a continuación se debe seleccionar la imagen que se pretende restaurar y por último seleccionar el método de restauración que se desea emplear (se ha comentado un poco antes los distintos métodos que se puede emplear). Finalmente, se ha de confirmar mediante el botón «Aceptar» para que se inicie el proceso.

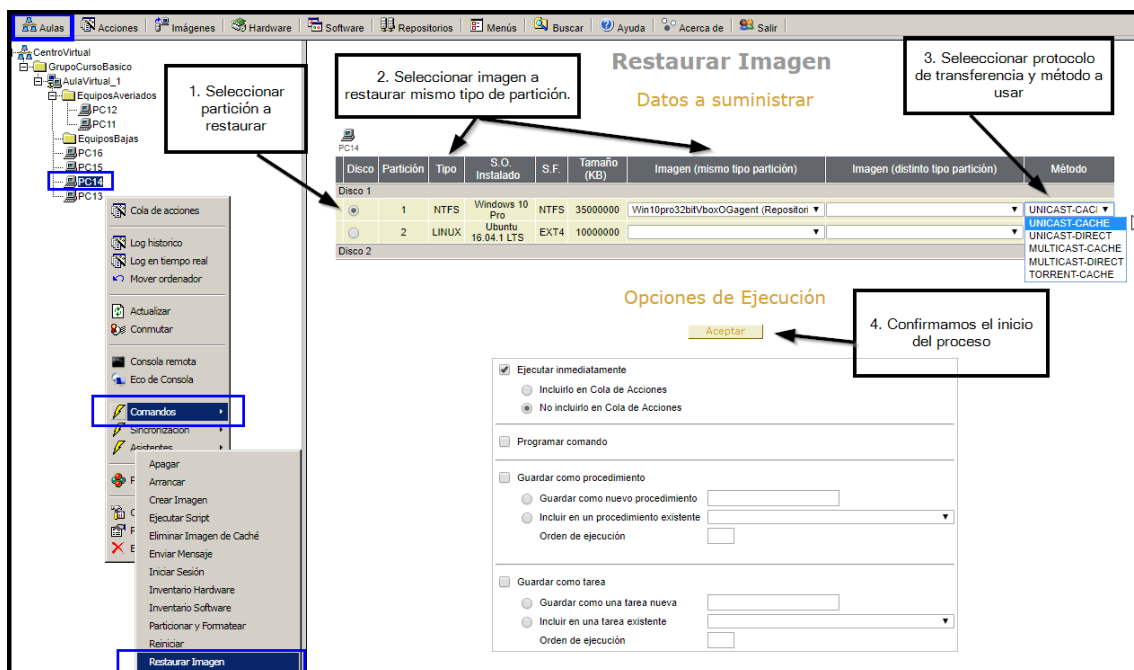


Figura 4.61: Comando de restaurar una imagen desde OpenGnsys

El comando «Restaurar» puede cambiar automáticamente el método de cache a direct y el protocolo a unicast para que la operación se realice correctamente, aunque esto puede causar que se tarde más tiempo en concluir el proceso; es decir, se puede realizar un despliegue con un método que utilice el repositorio local cache y si tenemos equipos en el destino que no cumplen los requisitos de despliegue, entonces automáticamente se cambia a unicast-direct en dichos equipos para que concluya el despliegue (con el consiguiente aumento de tiempo). Este comportamiento se puede configurar editando el archivo `/opt/opengnsys/client/etc/engine.cfg` en la línea que sigue:

```
#Que protocolo de restauracion usar en el caso de que no exista cache o no
  exista espacio suficiente. [NONE | unicast | multicast].NONE retorna error
RESTOREPROTOCOLNOTCACHE=NONE
```

4.10.2. Asistente Deploy de imágenes

El Asistente Deploy de la consola web de OpenGnsys implementa la actualización de la cache (updateCache).

Como se puede observar en la figura 4.62, lo primero que se hace es seleccionar sobre un ámbito de aplicación (en este caso se trata de un ordenador) el asistente de Deploy de imagen. Desde aquí y ya en el formulario lo primero que se ha de hacer (punto 1) un despliegue completo: actualizar la caché (updateCache) y la restauración desde el repositorio local cache a la partición de destino. También se puede seleccionar únicamente la actualización del repositorio local cache (esta opción es interesante en aquellos casos que se realice un proceso de actualizar el repositorio local de un ámbito de ordenadores por la noche -se dispondría de una copia actualizada de la imagen que se quiere restaurar localmente-).

Se ha de seleccionar unas opciones básicas (punto 2) necesarias para el proceso del Deploy de la imagen: se ha de seleccionar la partición del destino donde se va a realizar la restauración, seleccionar la imagen a distribuir y por último, el método de transferencia (las diferentes opciones se han explicado brevemente en la sección 4.10.1).

The screenshot displays the 'Asistente de Despliegue de la imagen' web interface. At the top, it shows the scope 'Ámbito: Ordenadores,11aim-15'. Below this is a table listing disks and their partitions:

Disco	Partición	Tipo	S.F.	S.O. Instalado	Tamaño (KB)	Imagen	Perfil Software	Fecha/Caché
Disco 1	1	EMPTY	EMPTY		0		Perfil Software (11aim-15, Part 1)	
	2	EMPTY	EMPTY		0			
	3	EMPTY	EMPTY		250000000			
	4	CACHE	CACHE		488386584			Caché libre: 233350.MB
Disco 2	1	NTFS	NTFS	Windows 10 Enterprise LTSC 2019 1809 64 bits	485000000		Perfil Software (11aim-15, Part 1)	2019-07-06 00:06:27
		MSDOS			500107608			

Below the table, there are three main sections:

- 1.- Opcional: Solicitar un updateCache**: A checkbox option for updating the cache.
- 2.- Selección del destino, el origen y del protocolo**: A section for selecting the target partition (Disco 2 - Part 1), the image to distribute (Imagen completa con programas 1 Julio 2019 (AIM_REPO)), and the transfer method (TORRENT).
- 3.- Opcional: personalizar opciones del protocolo**: A section for customizing protocol options, including multicast, torrent, and unicast settings.

At the bottom, there are buttons for 'Generar Instrucción OG' (labeled 4.- Generar el código a ejecutar) and 'Aceptar' (labeled 5.- Iniciar el proceso).

Figura 4.62: Asistente Deploy de imagen desde la consola web de OpenGnsys

En el punto 3 se muestra las opciones del protocolo que se ha elegido para la transferencia de la imagen (tanto para la parte de multicast como de torrent). Con todo ello, se debe presionar el botón «Generar Instrucción OG» (punto 4) para ver el código que se genera. Dicho código se puede guardar como un procedimiento para en un futuro utilizar el procedimiento en vez del asistente (además con la API de programación un administrador de la OU podría añadir algunos pasos más a la hora de realizar el Deploy). Finalmente, se ha de iniciar el proceso que se ha configurado previamente. Para ello se ha de presionar el botón «Aceptar» del formulario del asistente de despliegue de la

imagen (punto 5).

Una vez lanzado el deploy (ya sea mediante el comando «Restaurar» o por medio del asistente) se podrá observar en el «Estado de los ordenadores» sobre el ámbito de ejecución sobre el que se ha realizado este deploy, que los equipos se ponen de color rojo indicando que están ocupados. Conviene recordar que antes de lanzar el deploy, los equipos sobre los que se va a lanzar tienen que estar arrancados con el cliente PXE ogLive (para ello se ha de seleccionar con el menú contextual «NetBoot avanzado» para que arranquen del ogLive (tal como se ha explicado en la sección 4.6.3.6).

El proceso de restaurar una imagen creada previamente a unos equipos de destino suele tardar un tiempo importante (depende del tamaño de la imagen, la velocidad de transmisión, el estado de los equipos, etc.). En cualquier caso, es conveniente realizar el seguimiento de la operación a través de los logs que se generan en los clientes (tal como se explicó en la sección 4.9.5).

En la figura 4.63 se muestra un log en tiempo real en la creación de la imagen del sistema del equipo «L1aim-15». Muestra bastante información que va a ser muy útil para verificar los pasos que están realizando en la ejecución de un proceso.

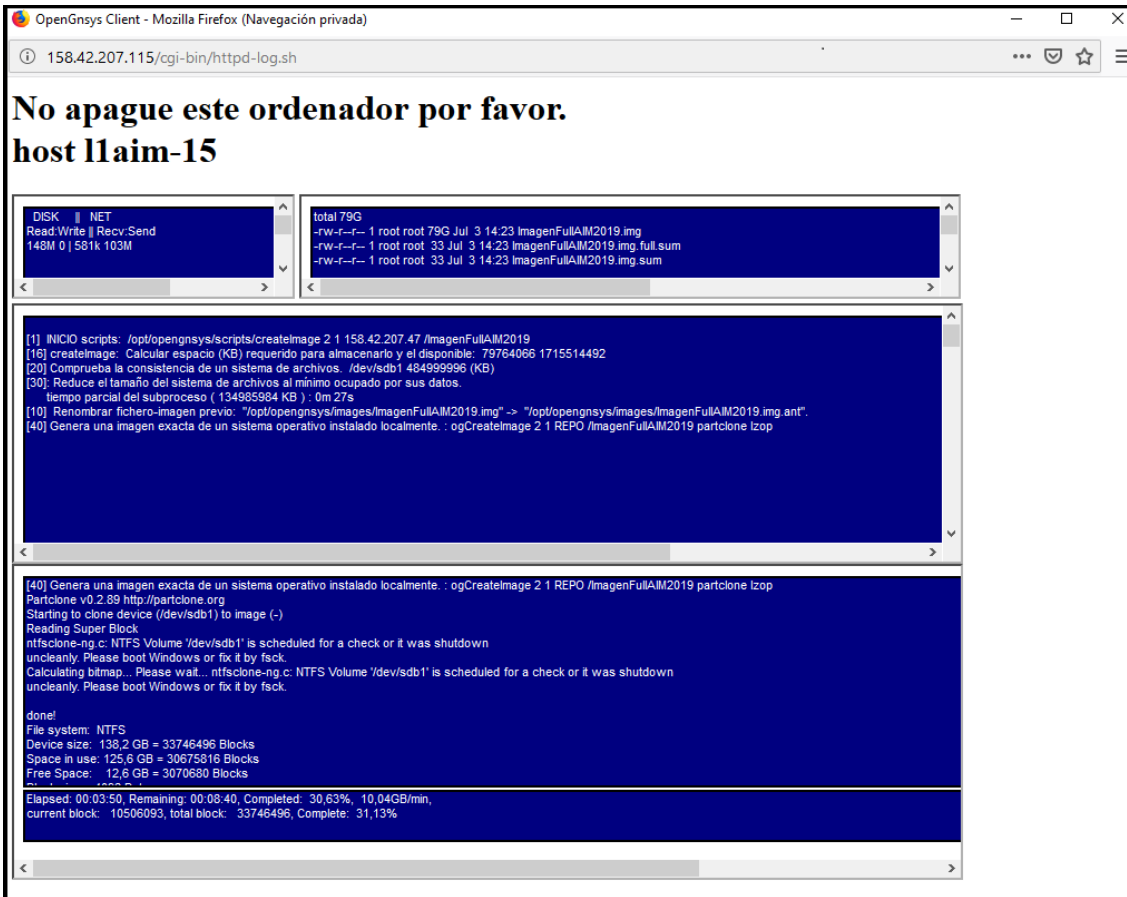


Figura 4.63: Log en tiempo real proceso creación de una imagen desde la consola web de OpenGnsys

En el caso de no haber podido observar el log en tiempo real, se puede hacer uso del log histórico tal como se puede mostrar en la figura 4.64. En el fragmento del log que se

ha mostrado se puede observar el inicio de una restauración de una imagen utilizando el protocolo unicast-direct sobre la primera partición del segundo disco del equipo. La imagen que se ha restaurado se llama «ImagenFullAIM2019». Al finalizar el proceso realiza una serie de acciones: deshabilitar el chequeo del disco, cambia el nombre al equipo, activa el controlador del disco, instala un mini setup, mete el equipo en el dominio, instala el antivirus, mueve el equipo al servidor de Kaspersky adecuado y finaliza la configuración. El tiempo total empleado fue de 12 minutos y 9 segundos.

```
[1] INICIO scripts: /opt/opengnsys/scripts/deployImage 158.42.207.47 ImagenFullAIM2019 2 1 UNICAST-DIRECT
Desmonta un sistema de archivos. 2 1
[1] REPO=158.42.207.47 IMG-FILE=ImagenFullAIM2019.img SIZE=75452328 (KB) METADATA=PARTCLONE:LZOP:NIFS:138200000
[55] Restaura una imagen de sistema operativo.: restoreImage 158.42.207.47 ImagenFullAIM2019 2 1
[1] INICIO scripts: /opt/opengnsys/scripts/restoreImage 158.42.207.47 ImagenFullAIM2019 2 1
[40] ogRestoreImage REPO ImagenFullAIM2019 2 1 UNICAST
[100] Duracion de la operacion 11m 16s
[ ] Total Time: 00:11:14, Ave. Rate: 11,2GB/min, 100.00% completed!
[ ] tiempo parcial del subprocesso : 11m 16s
[90] Iniciar la configuración del sistema restaurado
Deshabilita el CHKDSK
Cambia el nombre del equipo a llaim-14
Activar controladoras de disco
Instala Mini Setup
Se procede a meter el equipo en el dominio
Se ha metido el equipo en el dominio
Instala el antivirus
Finaliza la instalacion antivirus
Mover equipo al Servidor de Kaspersky
Finaliza configuracion Windows
[100] tiempo total del proceso 12m 9s
[END Interface] Comando terminado con este código: 0
```

Figura 4.64: Log histórico de un equipo desde la consola web de OpenGnsys

4.10.3. Postconfiguración de los sistemas operativos

Después de haber realizado una restauración o deploy del sistema de archivos, se han de configurar los clientes afectados antes de su primer inicio.

La postconfiguración básica incluye el cambio de nombre del equipo, la configuración o reinstalación del gestor de arranque y procesos para independizar el sistema restaurado del hardware.

El fichero donde se puede configurar los pasos necesarios que queremos hacer en los equipos se llama *configureOS* que se puede editar desde el servidor en la siguiente ruta: `/opt/opengnsys/client/scripts/configureOs`. En la figura 4.65 se muestra una serie de configuraciones que se han añadido a la hora de restaurar una imagen.


```

# Nombre del cliente.
HOST="$(ogGetHostname)"

# Post-configuración personalizada para cada tipo de sistema operativo.
OSTYPE="$(ogGetOsType $1 $2)"
case "$OSTYPE" in
  Windows) # Postconfiguración de Windows.
    # Asignar nombre del equipo.
    HOST=${HOST:-"pc"}

    # Deshabilita el CHKDSK después del arranque
    ogEcho log session "Deshabilita el CHKDSK"
    ogUnsetDirtyBit $1 $2
    ogEcho log session "Cambia el nombre del equipo a $HOST"
    ogSetWindowsName $1 $2 "$HOST"

    # Configurar el boot sector de la partición Windows.
    ogFixBootSector $1 $2
    # Configurar el gestor de arranque de Windows XP/Vista/7.
    ogWindowsBootParameters $1 $2
    # Registrar en Windows que la partición indicada es su nueva unidad C:\
    ogWindowsRegisterPartition $1 $2 C $1 $2

    # Deshabilitar el arranque rápido de Windows para evitar problemas tras el apagado
    #ogEcho log session "Deshabilita el FastBoot"
    #ogDisableFastBoot $1 $2

    #MODIFICACIONES TEMPORALES DEL CURSO BASICO OPENGNSYS
    ogEcho log session "Activar controladoras de disco"
    buildToOrder $1 $2
    #FIN MODIFICACIONES SECCION WINDOWS

    # Metemos en el dominio al equipo
    # Modificaciones Fede Olmeda
    DOMINIO=alumno
    USU=agregar
    PASS=CONTRESEÑA
    ADMIN=admin
    PASSADMIN=PASSWORD
    ogEcho log session "Instala Mini Setup"
    ogInstallMiniSetup $1 $2 postconf.cmd $ADMIN $PASSADMIN 0
    ogEcho log session "Se procede a meter el equipo en el dominio"
    ogDomainScript $1 $2 postconf.cmd $DOMINIO $USU $PASS
    ogEcho log session "Se ha metido el equipo en el dominio"
    ogInstallWindowsClient $1 $2 postconf.cmd
    ogEcho log session "Instala el antivirus"
    ogAddCmd $1 $2 postconf.cmd "msiexec /i c:\progra-2\kasper-1\agente\exec\kasper-1.msi /qn SERVERADDRESS=aimkasper.upvnet.upv.es EULA=1"
    ogEcho log session "finaliza la instalacion antivirus"
    ogEcho log session "Movemos el equipo al Servidor de Kaspersky"
    ogAddCmd $1 $2
    ogEcho log session "Finaliza configuracion Windows"
    # Fin de las modificaciones

```

Figura 4.65: Parte del fichero configureOs donde se ha realizado la postconfiguración de los clientes

4.11 Eliminar una imagen del sistema

Para eliminar una imagen del sistema se puede hacer desde una de las siguientes opciones:

- Desde la barra de herramientas «Repositorios».
- Desde la barra de herramientas «Imágenes».
- Manualmente desde la consola SSH.

Para eliminar una imagen del repositorio hay que ir a la pestaña «repositorio» de la barra de herramientas de la consola web de OpenGnsys y seleccionar del menú contextual del repositorio la opción «Eliminar Imagen Repositorio». Aparece un listado de imágenes del repositorio y se marca la que se quiere eliminar. Una vez se ha seleccionado el elemento que se desea borrar, se debe pulsar el botón aceptar.

En la figura 4.66 se observa que las imágenes normales son de tipo archivo (F) y las copias de seguridad de estas imágenes (B). Las imágenes de tipo *backup* son creadas automáticamente cuando se reutiliza un objeto de tipo imagen (el fichero imagen previo es renombrado a *.ant*).

Eliminar Imagen del Repositorio

Datos a suministrar: 158.42.207.47,AIM_REPO

Espacio TOTAL	Espacio Ocupado	Espacio Libre	% Ocupado
1.8T	168G	1.6T	10%

Imágenes Disponibles

Vista Unidad Organizativa

Nº	Marcar	Tipo	Nombre de la Imagen	Tamaño	Eliminar Objeto Imagen	Unidad Organizativa
1	<input type="checkbox"/>	F	ImagenFullAIM2019	72G	<input type="checkbox"/>	AIM
2	<input type="checkbox"/>	B	ImagenFullAIM2019.ant	79G	-----	
3	<input type="checkbox"/>	F	W10Base12062019	7.9G	<input type="checkbox"/>	AIM
4	<input type="checkbox"/>	B	W10Base12062019.ant	8.8G	-----	

Figura 4.66: Información de una imagen del repositorio que se pretende eliminar

También se puede borrar la definición de un objeto imagen. Para ello y tal como se puede ver en la figura 4.67, el administrador de una unidad organizativa debe seleccionar de la barra de herramientas la pestaña «Imágenes» y en el árbol de objetos se muestran las imágenes que hay. Seleccionamos la imagen que nos interesa eliminar y con el menú contextual «Eliminar Imagen» aparece el formulario de gestión de las imágenes monolíticas y presionar aceptar para confirmar la eliminación de la imagen seleccionada.

Gestión Imágenes Monolíticas

Eliminar

Nombre	W10Base12062019
Descripción	Imagen base SO aulas AIM
Repositorio	AIM_REPO
Comentarios	Imagen base W10 1903 x64 limpia 29/06/2019
Acceso remoto	<input type="checkbox"/>
Equipo modelo	l2aim-02 (Aula2)
Disco, Partición	1, 1 (0-EMPTY)
Fecha de creación	2019-07-02 19:22:09 (r10)
Perfil Software	Perfil Software (l2aim-02, Part.1)
Sistema operativo	Windows 10 Enterprise LTSC 2019 1809 64 bits
Datos del repositorio	
Camino	W10Base12062019.img
Tamaño	7.89 GB
Copia de seguridad	8.73 GB

ATENCIÓN.- Una imagen que tiene perfil software indica que en algún momento se ha creado y por tanto no pueden modificarse ciertos datos a menos que la elimine y la vuelva a crear

Figura 4.67: Eliminar un objeto imagen

El administrador de la OU también puede eliminar manualmente una imagen desde una consola SSH al servidor utilizando una cuenta con permisos para realizar estas operaciones. Este borrado debe realizarse de forma segura para evitar la pérdida de datos. Los ficheros de imágenes se almacenan en el directorio /opt/opengnsys/images de cada

repositorio.

Una imagen consta normalmente de tres ficheros:

- `NombreCanónico.img`: es la imagen que contiene el volcado completo del sistema de archivos y ocupa una gran cantidad de espacio en el disco donde se almacena.
- `NombreCanónico.img.sum`: suma parcial de comprobación de seguridad de la imagen.
- `NombreCanónico.img.full.sum`: suma completa de comprobación de seguridad de la imagen.
- `NombreCanónico.img.torrent`: datos para restaurar la imagen usando el protocolo P2P.

Es posible que tengamos también algunos archivos con extensión `.ant` que contienen una copia de seguridad de la imagen.

Para borrar los ficheros asociados a una imagen, hay que introducir desde la consola SSH como parámetro el nombre canónico de dicha imagen mediante el siguiente comando:

```
sudo deleteimage NombreCanónico
```

Mediante el siguiente comando se eliminan los ficheros de la imagen normal y de su copia de seguridad (con la extensión `.ant`):

```
sudo deleteimage -b NombreCanónico
```

Para recuperar la copia de seguridad renombrando automáticamente los ficheros para que se puedan utilizar como una imagen normal, se emplea el siguiente comando⁵:

```
sudo deleteimage -r NombreCanónico
```

4.11.1. Eliminar la imagen de cache en el cliente

Los ordenadores clientes pueden disponer de una partición de tipo cache para alojar imágenes entre otras cosas. Desde la consola web se puede ver el tamaño (en KB) de la partición cache y del espacio libre. Para ello se ha de seleccionar el equipo sobre el que se quiere consultar el estado de la cache y seleccionar del menú contextual «configuración».

Para poder eliminar una imagen del repositorio local cache se precisa que los equipos clientes estén iniciados en ogLive. Desde la consola web de OpenGnsys se selecciona el ámbito de ejecución que nos interese utilizar y utilizamos del menú contextual del objeto seleccionado el comando «Eliminar Imagen de cache». En este momento se debe mostrar el formulario de Eliminar Imagen de la cache en cliente tal como se puede observar en la figura 4.68. Se puede observar que se ha seleccionado todo el Aula1 y con el menú contextual se ha seleccionado el comando adecuado. En el formulario aparece la imagen que se puede borrar del repositorio local cache de los equipos que pertenecen al ámbito de Aula1. Se debe presionar sobre el botón aceptar para que se aplique la eliminación.

⁵Los parámetros `-b` y `-r` son opcionales y mutuamente excluyentes

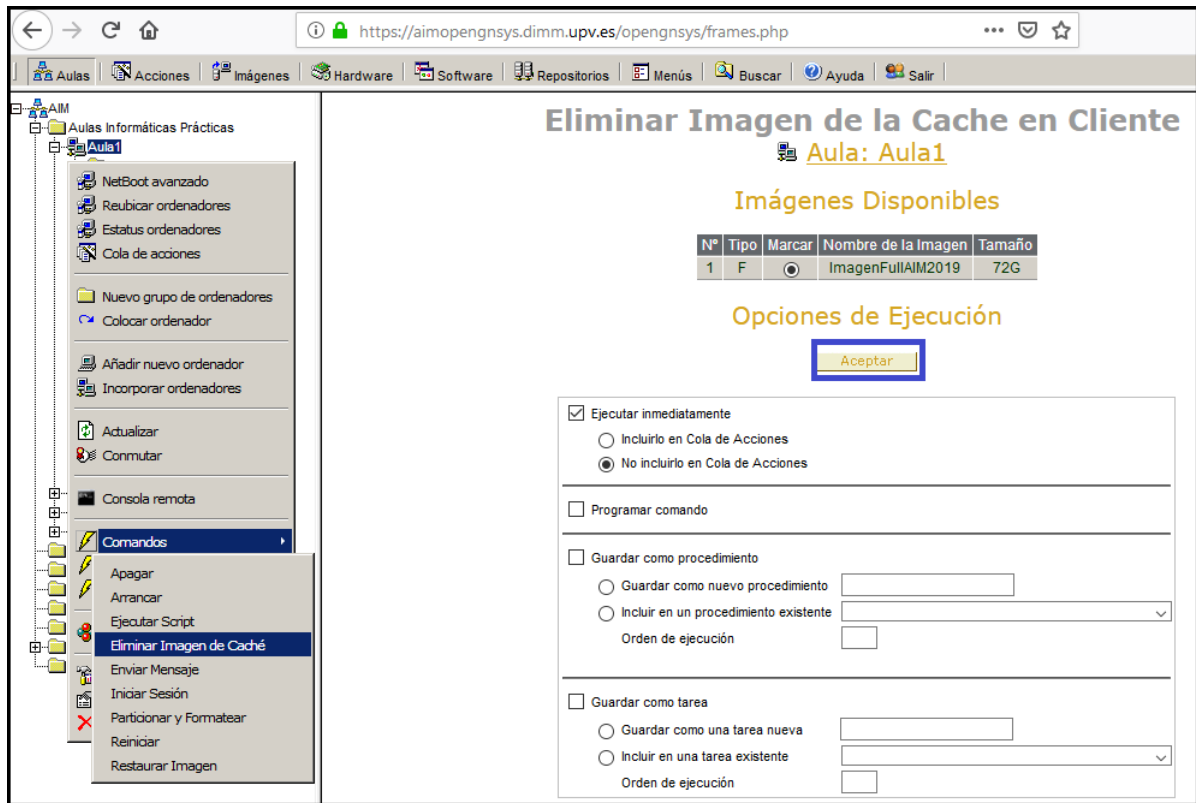


Figura 4.68: Eliminar imagen de la caché en cliente desde la consola web de OpenGnsys

4.12 Funcionamiento global de OpenGnsys

Una vez tenemos claro el funcionamiento de la aplicación para la realización del clonado de equipos en un aula, departamento, centro universitario o empresa, voy a resumir brevemente los pasos que se han de seguir (se parte de un servidor OpenGnsys instalado y con su repositorio tal como se ha visto en las secciones 4.1 y 4.2) :

1. Establecer el esquema lógico de nuestra organización por medio de la consola de administración de OpenGnsys (se debe crear una o varias unidades organizativas con su delegación a usuarios para su administración).
2. Definir nuestro repositorio que va a albergar las imágenes que se van a necesitar.
3. Configurar el esquema lógico en la unidad organizativa creada (grupo de aulas, aulas, grupo de ordenadores y ordenadores). Con esta parte se definen los ámbitos de ejecución.
4. Creación de las imágenes que vayamos a necesitar para nuestra organización.
5. Coger un equipo que va a servir de equipo modelo y particionarlo adecuadamente. Instalar el sistema operativo que se vaya a emplear y el resto del software necesario. Configurar adecuadamente el sistema hasta que estemos conforme con las necesidades.
6. Crear una imagen del equipo modelo que nos va a servir de referencia.
7. Configurar el archivo configureOS para que tras la restauración en los clientes se aplique las configuraciones que se han creído convenientes.

8. Crear un deploy a los clientes sobre los que queremos actuar. Previamente se habrá particionado adecuadamente. Tras la restauración de la imagen se aplicará las configuraciones que se han definido en el fichero configureOS.
9. Verificar que todo ha sido correcto.
10. Seleccionar el tipo de arranque desde el menú contextual «Netboot avanzado».

Estos pasos serían la base en el procedimiento de gestión de los equipos. Evidentemente se tendrá que utilizar las herramientas que OpenGnsys nos proporciona para afinar los procesos que empleemos: encendido masivo equipos, apagado de los mismos, logs, etc.. Además disponemos de la consola web de OpenGnsys que nos permite visualizar de un modo muy sencillo cómo están los procesos en cada momento: qué equipos han arrancado, el tipo de sistema operativo que está iniciado, si las sesiones se han iniciado o no, etc.. Además de la herramienta gráfica, se dispone también de la consola SSH al servidor OpenGnsys que permite configurar una serie de funciones que desde la parte gráfica no es tan accesible o no se puede. No solamente podemos conectarnos al servidor OpenGnsys; también se puede realizar una conexión SSH a los equipos clientes y podemos acceder a sus configuraciones y sus discos (para ello es necesario que el cliente haya arrancado el sistema operativo ogLive).

En definitiva, se trata de una herramienta muy útil y potente que un administrador de un departamento u organización gestione sus equipos frente a fallos del sistema, mala configuración de los sistemas, errores físicos y de software, reemplazo de discos mayores, etc.. A pesar del trabajo dedicado en el aprendizaje y puesta en marcha en los espacios del AIM, estoy convencido que se le puede sacar mayor provecho a esta herramienta para el óptimo funcionamiento de los equipos que están bajo mi administración. Se pretende mejorar los procedimientos y técnicas de clonado de los equipos de tal manera que se puedan automatizar muchos procesos.

CAPÍTULO 5

Herramientas para la automatización de procesos en los equipos informáticos

Un administrador de sistemas necesita realizar muchas operaciones cuya finalidad es buen funcionamiento de los equipos que gestiona. Muchas de las tareas que se tienen que supervisar se pueden automatizar y poder despreocupar de esas tareas al administrador.

Según [43], un *script* es un programa usualmente simple que se suele almacenar en un archivo de texto plano. Generalmente los *scripts* son casi siempre interpretados y se suele emplear para diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario.

Nos podríamos preguntar la diferencia de un lenguaje compilado de uno interpretado. El lenguaje compilado requiere un paso adicional antes de ser ejecutado, la compilación, que convierte el código que escribes a un lenguaje de máquina (instrucciones que entiende el computador). Sin embargo, un lenguaje interpretado, es convertido a lenguaje de máquina a medida que es ejecutado.

Las ventajas del lenguaje compilado son:

- Tiempos de ejecución rápidos.
- Código fuente inaccesible.
- Están preparados para ejecutarse.

Con respecto a los inconvenientes se tiene:

- Son poco flexibles.
- No son multiplataforma.
- Requieren un paso extra (compilación).

Como ejemplos de lenguajes compilados tenemos: JavaScript, C, C++, Java, etc..

Las ventajas del lenguaje interpretado son:

- Multiplataforma. Es independiente de la máquina y del sistema operativo ya que no contiene instrucciones propias de un procesador sino que contiene llamadas a funciones que el intérprete reconoce.
- Permite modificar en tiempo de ejecución el código que se está ejecutando.
- Son sencillos de probar.

Con respecto a los inconvenientes se tiene:

- Velocidad de ejecución es más lenta.
- Portabilidad. Se requiere un intérprete.
- El código fuente es público.

Ejemplos de lenguajes interpretados: PowerShell, PHP, ASP, Perl, TCL, Bash, Python, etc..

Hasta relativamente hace poco, **command.com** y **cmd.exe** eran los intérpretes de líneas de comandos por defecto en los sistemas operativos DOS y Windows, permitiendo a los usuarios avanzados abrir aplicaciones de consola, solucionar problemas o navegar por las unidades de los discos duros de un equipo. Sin embargo, estos shells han tenido dos problemas:

1. No tienen acceso a todos los componentes del sistema.
2. Sus lenguajes de *script* se consideran limitados en su funcionalidad.

Es por estos problemas o inconvenientes que siempre se han considerado menos eficientes y completos que los shells de Linux y Unix actuales. Con el fin de eliminar estos problemas, Microsoft creó la herramienta PowerShell como una solución central de administración y automatización en Windows.

Entre las características de PowerShell, se tienen:

- Mantiene las características del cmd.exe.
- Es una shell de línea de comandos y un lenguaje de secuencia de comandos.
- Todos los comandos están disponibles desde la línea de comandos.
- Puede interactuar con un montón de tecnologías: .NET Framework, registro de Windows, COM, WMI, ADSI. Exchange, Sharepoint, System Center, Hyper-V, SQL, etc..
- Funcionamiento más eficiente gracias a una estructura de comandos y una sintaxis coherentes.
- Está basado en objetos y no en cadenas como se tenía anteriormente. Cada objeto tiene propiedades y métodos que definen cómo se puede utilizar. Permite la ejecución de tareas más complejas mediante tuberías (pipelining).
- Es un completo entorno de scripting que permite una automatización eficiente de los procesos.

Respecto a los inconvenientes de PowerShell, se tiene:

-
- Dirigido principalmente a profesionales de TI y administradores de sistemas.
 - No tiene ningún beneficio adicional para los usuarios acostumbrados a una GUI ya que raramente usan la línea de comandos.
 - Cuesta un poco acostumbrarse a su funcionamiento.

Además de todo lo comentando, se ha de tener en cuenta lo siguiente:

- Bajo entornos Windows, todo lo que ejecutas en la interfaz gráfica corre por debajo con PowerShell.
- Todos los productos creados por Microsoft utilizan PowerShell y en muchas de estos de estos productos existen procedimientos que no se pueden realizar a través de una interfaz gráfica y solo se puede realizar con PowerShell.

Teniendo en cuenta que los equipos que se emplean en el AIM utilizan sistemas operativos Windows (en sus versiones cliente y servidor), PowerShell es realmente una opción ganadora por los siguientes motivos:

- VBScript está obsoleto, y su desarrollo se detuvo hace mucho.
- PowerShell se ha acercado considerablemente a las capacidades de VBScript, Python, Perl, etc. que son muy potentes en Linux pero que son menos útiles en Windows debido a no pueden acceder a muchos y variados usos de las APIs de Windows para su gestión.
- Microsoft está realizando un gran esfuerzo en recursos para que PowerShell sea una herramienta mejor y más potente, que además dispone de una comunidad global muy involucrada.

Por todo ello, y sin menospreciar el resto de herramientas que son muy potentes y con las que se pueden realizar también auténticas maravillas en la administración de sistemas, se ha decidido la utilización de PowerShell. No se ha realizado un estudio riguroso de las distintas herramientas que se pueden emplean para la automatización y gestión de los procesos en los equipos informáticos que se han de administrar ya que el objetivo en este trabajo no es seleccionar la mejor y más completa herramienta de las diferentes alternativas que se pueden encontrar en la actualidad. PowerShell cumple sobradamente las necesidades que se han propuesto para este trabajo.

Como se comentó en la sección 1.1, en este trabajo no se pretende realizar una solución completa (las hay en el mercado) para la administración de sistemas sino un inicio hacia una administración mejor, libre de errores no deseados, ágil y sencilla, etc.. PowerShell es muy potente y el límite lo pone el administrador del sistema o la limitación en sus prácticas y conocimientos.

CAPÍTULO 6

Introducción al scripting

Todos los sistemas operativos tienen una Shell que permiten un trabajo más técnico o avanzado sobre las múltiples funciones, opciones y capacidades del mismo. En los sistemas Windows tendríamos los programas "Command.com" o "cmd.exe" y PowerShell. En los sistemas Unix existen varias implementaciones de la Shell como "Bourne Shell" y "C Shell" entre otras muchas.

La Shell es el programa que permite a los usuarios interactuar con el sistema operativo mediante el procesamiento de órdenes de texto en modo terminal y el Shell scripting es un nivel superior del manejo de las implementaciones de la Shell que facilita el manejo avanzado de las capacidades de los sistemas operativos y mejorar la automatización y la monitorización de tareas y procesos. Además de ofrecer la posibilidad de acceder a las funciones del sistema operativo permiten encadenar comandos que ejecutan tareas sencillas para así implementar tareas de cierta complejidad automatizando procesos que se suelen realizar con frecuencia y de una manera desatendida. Las características de los *scripts* de Shell son:

- Son fáciles de programar.
- Tienen un mayor costo de procesamiento cuando son ejecutados.
- Usan intérpretes en lugar de compiladores para ejecutarse.
- Pueden interactuar o comunicarse con componentes escritos en otros lenguajes de programación.
- Están escritos en archivos de texto sin formato.
- Poseen una estructura de código muy pequeña en comparación con otros programas similares realizados en otros lenguajes de programación compilados.

Hay veces a que una Shell se le llama también intérprete de comandos (IC). Un intérprete de comandos es un mero programa que recibe unos datos de entrada, los procesa y devuelve unos resultado como salida. Sirve de interfaz entre el usuario y el sistema operativo ya que los ordenadores no entienden nuestro lenguaje y se necesita un programa intermedio que cuando introduzcamos alguna orden el ordenador sea capaz de entenderlo. Si un ordenador puede ser controlado mediante aplicaciones fáciles de usar que tienen una interfaz gráfica podríamos preguntarnos el por qué introducir comandos a través de la línea de comandos. Las razones principales de ello son:

- Automatización de los comandos.

- Acceso directo a las funciones del sistema operativo.
- Requiere pocos recursos.

Hoy en día se puede elegir entre muchos intérpretes de comandos. A la hora de elegir uno, se debe tener en cuenta criterios como facilidad de uso, potencia o portabilidad. A pesar de que los lenguajes interpretados son más lentos que los lenguajes compilados permiten desarrollos muy rápidos y se suelen emplear en la administración de sistemas operativos.

En el día a día de un administrador de sistemas se realizan multitud de tareas repetitivas. Es por ello que para facilitar dicha tarea se emplea el uso de los *scripts* cuyo objetivo es automatizar y encadenar tareas relacionadas con los sistemas. Se podría definir un *script* como un conjunto de órdenes guardadas en un archivo de texto (suelen ser archivos muy ligeros) y que es ejecutado por lotes o línea a línea en tiempo real por un intérprete que codifica la información del *script* y lo transforma en lenguaje máquina para ser procesada por el ordenador.

Se recomienda que a la hora de crear *scripts* de manera ordenada y estandarizada de tal manera que sea fácilmente entendible con muy poco esfuerzo en un futuro por su creador u otros administradores. Es fundamental documentar dichos códigos para evitar la ardua y tediosa tarea de averiguar qué codificó, cómo y por qué y también para saber por qué en un futuro deja de funcionar el código. Entre las prácticas recomendadas se tiene:

- Indentar el código. Disponer un código legible es muy importante por lo que se recomienda indentar adecuadamente el código para tener una estructura lógica adecuada.
- Añadir espacios de separación entre secciones de código.
- Comentar adecuadamente el código.
- Crear variables con nombres descriptivos de sus funciones.

Teniendo en cuenta todo lo explicado anteriormente, nos podríamos preguntar el por qué del *scripting* y del uso de la Shell. El hecho de ganar tiempo es sin duda un factor importante a la hora de usar una secuencia de comandos y de la automatización de procesos. Entre otras ventajas de ello se tiene las siguientes:

- Consistencia. Utilizar un *script* implica que se va a ejecutar siempre igual evitando errores tipográficos, completándose la tarea y reduciendo el error humano.
- Auditoría. Hay muchas tareas en las que es útil tener un seguimiento de la misma: si se realizó la tarea, obtención de resultados importantes, saber los errores que ocurrieron, cuando se inició, quién inició la tarea, etc..
- Código modular.
- Documentación. Es importante que la secuencia de comandos que se empleen esté bien comentada y se documente el código con el fin de tener mejores conocimientos. Esta tarea se puede realizar de manera manual y en ciertos *scripts* de manera automática.
-

CAPÍTULO 7

PowerShell

Para el desarrollo del capítulo de PowerShell se ha tenido en cuenta algunos libros que han servido de consulta. En concreto los libros [44], [45] y [46].

7.1 ¿Qué es PowerShell?

Existe una definición en Wikipedia [47] que define perfectamente qué es Windows PowerShell: "Es un Framework desarrollado por Microsoft que nos permite automatizar tareas y gestionar configuraciones. Consiste en una línea de comandos y en un lenguaje de scripting asociado que está basado en .Net Framework".

En la tabla 7.1 se puede ver las distintas versiones de PowerShell que han ido apareciendo. Se puede observar que las versiones de PowerShell vienen ligadas con los sistemas operativos de Microsoft tanto en la parte cliente como en la parte servidora; es por ello, que cuando Microsoft saca una nueva versión de sus sistemas operativos está sacando también una nueva versión de PowerShell.

Versión	Fecha de lanzamiento	Sistemas Operativos por defecto
PowerShell 1.0	Noviembre 2006	Windows Server 2008 ¹
PowerShell 2.0	Octubre 2009	Windows 7 Windows Server 2008 R2 ²
PowerShell 3.0	Septiembre 2012	Windows 8 Windows Server 2012
PowerShell 4.0	Octubre 2013	Windows 8.1 Windows Server 2012 R2
PowerShell 5.0	Febrero 2016	Windows 10
PowerShell 5.1	Enero 2017	Windows 10 actualización aniversario Windows Server 2016

Tabla 7.1: Versiones de PowerShell

Las versiones de PowerShell soportan compatibilidad hacia atrás con el resto de versiones; sin embargo, cada versión incorpora nuevos cmdlets que no se pueden ejecutar en versiones posteriores. Es por ello que se recomienda actualizar clientes y servidores a la última versión de PowerShell.

¹Se instala mediante Server Manager

²Integrable con las versiones superiores de PowerShell

Como se ha indicado al inicio de esta sección, PowerShell es un intérprete de comandos que está orientado a la administración del sistema de los productos de Microsoft. Mediante PowerShell se puede automatizar un conjunto de tareas sobre un número indefinido de puestos de trabajo.

PowerShell puede ejecutar básicamente tres tipos de comandos individuales:

- **cmdlets.** Son programas escritos en .NET y pensados específicamente para trabajar con PowerShell.
- **Funciones PowerShell.**
- **Scripts de PowerShell (.ps1).** Son parecidos a los archivos .bat y se ejecutan conjuntos más o menos complejos de comandos para realizar tareas que se necesiten.

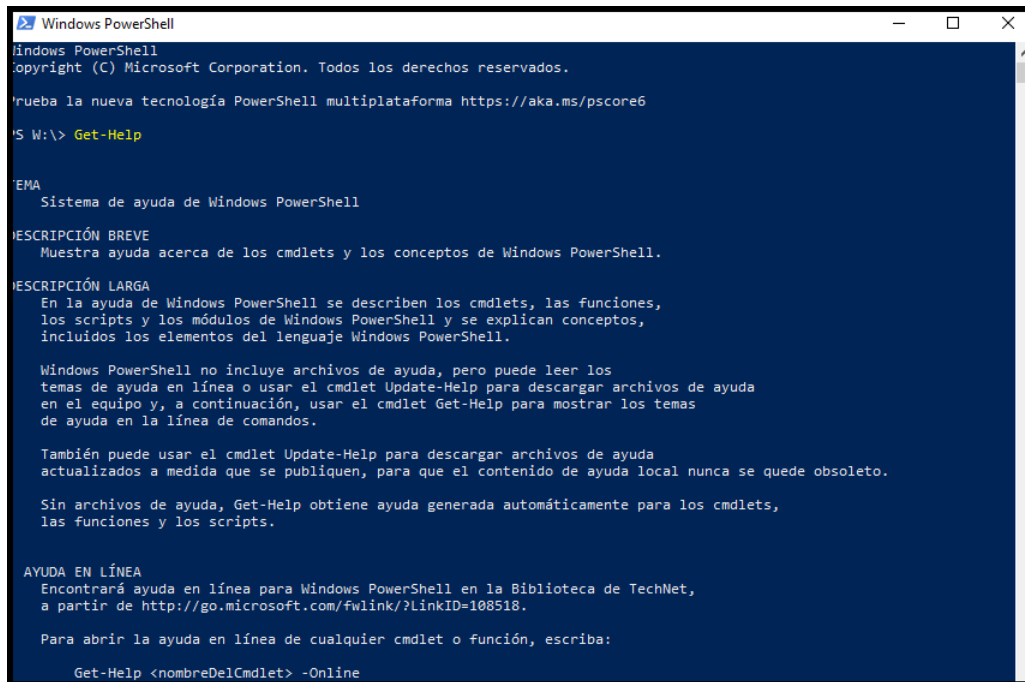
Unos de los mayores problemas que tienen los administradores de sistemas tradicionalmente es que no les gusta en exceso programar y los programadores no les gusta administrar sistemas. Sin embargo, PowerShell es una estupenda forma de aunar ambos mundos (programación y administración) y conseguir una potencia enorme bajo entorno Windows.

7.1.1. PowerShell v.5.1

Esta versión viene instalada de manera predeterminada en Windows 10 y Windows Server 2016. Entre sus muchas características se podría destacar las siguientes:

- Gestión de todos los roles y características de Windows.
- Soporte nativo para tareas programadas.
- Conectividad robusta que permite que sesiones remotas se mantengan incluso cuando ocurren fallos en la red.
- PowerShell Direct.
- Los usuarios pueden desconectar y reconectar sesiones sin perder su estado.
- Más fácil de aprender, usar y crear nuevos *scripts* (mediante PowerShell ISE).
- Windows PowerShell Web Access.
- Ayuda mejorada y actualizable.

Se dispone de varias consolas para trabajar con PowerShell. La primera de ellas es el PowerShell Cli -es el intérprete de línea de comandos (en inglés, command-line interpreter Cli)- tal como se puede observar en la figura 7.1. Se trata de un interface de línea de comandos que nos permite crear, editar y ejecutar instrucciones y *scripts* de PowerShell. Se incluye en la instalación de Windows por defecto desde Windows 7 y Windows Server 2008 R2.



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

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS W:\> Get-Help

EMA
    Sistema de ayuda de Windows PowerShell

DESCRIPCIÓN BREVE
    Muestra ayuda acerca de los cmdlets y los conceptos de Windows PowerShell.

DESCRIPCIÓN LARGA
    En la ayuda de Windows PowerShell se describen los cmdlets, las funciones,
    los scripts y los módulos de Windows PowerShell y se explican conceptos,
    incluidos los elementos del lenguaje Windows PowerShell.

    Windows PowerShell no incluye archivos de ayuda, pero puede leer los
    temas de ayuda en línea o usar el cmdlet Update-Help para descargar archivos de ayuda
    en el equipo y, a continuación, usar el cmdlet Get-Help para mostrar los temas
    de ayuda en la línea de comandos.

    También puede usar el cmdlet Update-Help para descargar archivos de ayuda
    actualizados a medida que se publiquen, para que el contenido de ayuda local nunca se quede obsoleto.

    Sin archivos de ayuda, Get-Help obtiene ayuda generada automáticamente para los cmdlets,
    las funciones y los scripts.

AYUDA EN LÍNEA
    Encontrará ayuda en línea para Windows PowerShell en la Biblioteca de TechNet,
    a partir de http://go.microsoft.com/fwlink/?LinkID=108518.

    Para abrir la ayuda en línea de cualquier cmdlet o función, escriba:

    Get-Help <nombreDelCmdlet> -Online
```

Figura 7.1: Consola PowerShell Cli

La siguiente consola es PowerShell ISE que se puede observar en la figura 7.2. Se trata de una aplicación con interfaz gráfica que está especialmente diseñada para crear, testear, ejecutar y depurar *scripts* de PowerShell. El uso de esta herramienta y los módulos de ayuda que dispone facilitan en gran medida la creación de código que se va a ejecutar posteriormente. Dentro de las muchas características que se agregaron al ISE cabe destacar "Intellisense" que completa automáticamente sus comandos mostrando menús de cmdlets, parámetros, valores de parámetros, archivos o carpetas coincidentes a medida que se escribe.

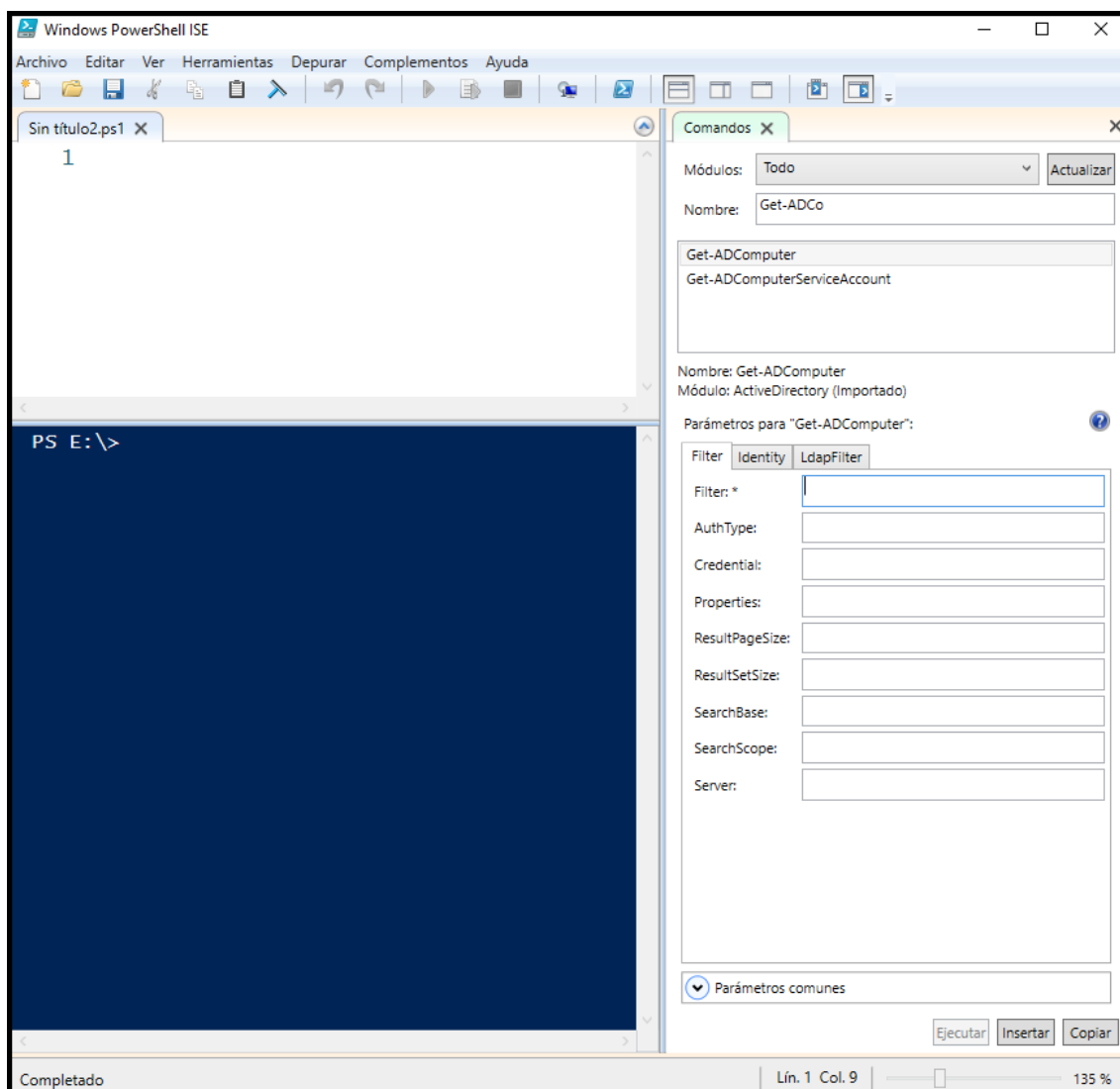


Figura 7.2: Consola PowerShell ISE

Cuando ponemos un comando desde la consola de PowerShell ISE, nos muestra las opciones que debemos ejecutar en el mismo (tanto las opciones obligatorias como las opcionales). De esta manera facilita mucho a la hora de incluir comandos en nuestro código.

Otra manera de acceder a PowerShell es a través de PowerShell Web Access. Se trata de una interfaz web que permite publicar PowerShell en el una página web (sería extender una consola Cli a una página web). Para ello habría que introducir nuestras credenciales en la página web y poder ejecutar comandos PowerShell en nuestros servidores que se hayan configurado. Para su instalación habría que ir a la documentación que se muestra en [48].

PowerShell Direct es una nueva funcionalidad de PowerShell que se incluye en Windows 10 y Windows Server 2016. Mediante esta funcionalidad nos permite conectarnos a las máquinas virtuales de Hyper-V y ejecutar comandos en las mismas sin necesidad de tener conexión de red. PowerShell desde versiones anteriores ya incorpora una funcionalidad (PowerShellRemote) que nos permite conectarnos a una máquina virtual remotamente y ejecutar comandos en la misma. Ahora esta funcionalidad se ha extendido un poco con PowerShell Direct de tal manera que si tenemos una máquina con Windows 10

con Hyper-V instalado, y queremos conectarnos a una máquinas virtuales se puede hacer usando Microsoft Virtual Machine Bus (VMBus) [49] en vez de usar las conexiones de red.

Una de las cosas más interesantes de PowerShell es poder lanzar ejecuciones en un equipo remoto. Existe un comando `Enter-PSSession` que permite conectar a un equipo remoto. Tal como se puede observar en la figura 7.3, se está intentando realizar una conexión a un equipo del área. Aparece un error de conexión de acceso denegado y nos insta a consultar la ayuda de `about_Remote_Troubleshooting`. A través de la consulta de dicha ayuda se puede tener una idea de los motivos por los que no se puede establecer la conexión a un equipo remoto.

```
PS D:\feolde\6_TFG\1_Scripts\Creaciones> Enter-PSSession masteraim-02
Enter-PSSession : Error de conexión al servidor remoto masteraim-02. Mensaje de error: WinRM no puede completar la
operación. Compruebe que el nombre del equipo es válido, que se puede acceder al equipo a través de la red y que la
excepción del firewall del servicio winRM está habilitada y permite el acceso desde este equipo. De manera
predeterminada, la excepción de firewall de WinRM para perfiles públicos limita el acceso a equipos remotos dentro
de la misma subred local. Para obtener más información, consulte el tema de la Ayuda about_Remote_Troubleshooting.
En línea: 1 Carácter: 1
+ Enter-PSSession masteraim-02
+ ~~~~~
+ CategoryInfo          : InvalidArgument: (masteraim-02:String) [Enter-PSSession], PSRemotingTransportException
+ FullyQualifiedErrorId : CreateRemoteRunspaceFailed
```

Figura 7.3: Intento de conexión a un equipo remoto

Se ha realizado un pequeño *script* que permite establecer la conexión a un equipo remoto utilizando para ello la herramienta de SysInternals llamada “psexec”. Dicho código se muestra en el listado de código 7.1

```
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Mediante este script se permite el acceso remoto a un equipo mediante
    PS.
    Para ello se utiliza la herramienta de Sysinternals "PSEXEC"
    que se conecta a un equipo remoto y se le lanza el comando winrm
    quickconfig.

.EXAMPLE
    .\Habilitar-PSRemoto.ps1 -NombreEquipo Equipo1 -RutaPsExec E:\Psexec.
    exe
.PARAMETER NombreEquipo
    Equipo sobre el que queremos activar la conexión remota
.PARAMETER RutaPsExec
    Path donde se encuentra el ejecutable psExec.exe
#>
[CmdletBinding()]
param (
    [Parameter(Mandatory = $True,
                ValueFromPipeline = $True,
                ValueFromPipelineByPropertyName = $True)]
    [string]$NombreEquipo,
    [Parameter(Mandatory = $False,
                ValueFromPipeline = $False,
                ValueFromPipelineByPropertyName = $False)]
    [string]$RutaPsExec = 'E:\PsExec.exe'
)
begin {
    function Test-PsRemoting {
```

```

    param (
        [Parameter(Mandatory = $true)]
        $NombreEquipo
    )
    try {
        $errorActionPreference = "Stop"
        $result = Invoke-Command -ComputerName $NombreEquipo { 1 }
    } catch {
        Write-Verbose $_
        return $false
    }
    if ($result -ne 1) {
        Write-Verbose "El acceso remoto a $NombreEquipo ha devuelto un
            resultado inesperado."
        return $false
    }
    $true
}

if (!(Test-Connection $NombreEquipo -Quiet -Count 1)) {
    throw "El equipo $NombreEquipo no se encuentra disponible"
} elseif (!(Test-Path $RutaPsExec)) {
    throw 'No se encuentra la aplicacion Psexec.exe'
}
}

process {
    if (Test-PsRemoting $NombreEquipo) {
        Write-Warning "El acceso remoto a $NombreEquipo se ha establecido"
    } else {
        Write-Verbose "Intentando el acceso remoto a $NombreEquipo..."
        & $RutaPsExec "\\$NombreEquipo" -s c:\windows\system32\winrm.cmd
        quickconfig -quiet 2>&1> $null
        if (!(Test-PsRemoting $NombreEquipo)) {
            Write-Warning "No se ha podido establecer acceso remoto con el
                equipo $NombreEquipo"
        } else {
            Write-Verbose "Se ha establecido acceso remoto con el equipo
                $NombreEquipo"
        }
    }
}
end {
}
}

```

Listado código 7.1: Código Habilitar-PSRemoto.ps1

Una vez lanzado el *script* "Habilitar-PSRemoto.ps1" que se ha mostrado en el listado de código 7.1 -no es necesario lanzarlo con privilegios de administración- y transcurrido un tiempo que tarda en realizar la operación, se está en condiciones de poder establecer una conexión remota con el equipo sobre el que se ha actuado, tal como se puede observar en la figura 7.4. Con la conexión establecida, se puede lanzar comandos de PowerShell sobre dicho equipo.

```
PS D:\feolde\6_TFG\1_Scripts\Creaciones> .\Habilitar-PSRemoto.ps1 masteraim-02
ADVERTENCIA: El acceso remoto a masteraim-02 se ha establecido

PS D:\feolde\6_TFG\1_Scripts\Creaciones> Enter-PSSession masteraim-02

[masteraim-02]: PS C:\Users\feolde\Documents>
```

Figura 7.4: Comandos para habilitar la conexión remota a un equipo

Llegados a este punto en el que tenemos una idea de lo que es PowerShell y de la importancia de automatizar procesos que se utilizan con frecuencia en el día a día, doy paso al siguiente apartado donde se explican las tareas que se han realizado en este trabajo.

7.1.2. Administración de un equipo en Windows

En esta sección se va a mostrar cómo realizar tareas administrativas que afectan directamente al núcleo del sistema operativo: habilitar y deshabilitar características de Windows, administración de los procesos, de los servicios, etc..

7.1.2.1. Características de Windows

Mediante el ejecutable DISM.exe lanzado desde una consola de comandos se puede instalar actualizaciones, paquetes de idiomas, características de Windows y mucho más. Además de ser un ejecutable está disponible como un módulo en Windows PowerShell (lo que significa que las acciones que se pueden realizar con DISM.exe tiene su equivalente con varios cmdlets).

Mediante la siguiente instrucción se puede consultar los cmdlets asociados al módulo DISM.exe (el listado es bastante grande por lo que no se va a mostrar).

```
Get-Command -Module dism
```

Para un administrador puede ser útil el poder activar una característica de Windows. Para poder hacerlo se ha de especificar el nombre de la característica mediante el parámetro **-FeatureName** del cmdlet **Enable-WindowsOptionalFeature**.

En la figura 7.5 se muestra cómo poder activar Internet Explorer 11 en un equipo con Windows 10. Esta configuración se hace localmente pero se puede realizar perfectamente en equipos remotos de una manera muy sencilla. Únicamente hay que tener en cuenta que se necesita lanzar la consola de PowerShell con privilegios elevados.

```
PS C:\Windows\system32> Enable-WindowsOptionalFeature -FeatureName Internet-Explorer-Optional-amd64 -Online

Path           :
Online         : True
RestartNeeded  : True
```

Figura 7.5: Activación de la característica de Windows 10 de Internet Explorer

Al realizar la activación de una característica de Windows, al finalizar solicita que reiniciemos el equipo.

También puede ser muy interesante para un administrador el desactivar una característica de Windows. Para poder hacerlo se ha de especificar el nombre de la característica

mediante el parámetro **-FeatureName** del cmdlet **Disable-WindowsOptionalFeature**, tal como se puede observar en la figura 7.6.

```
PS C:\Windows\system32> Disable-WindowsOptionalFeature -FeatureName Internet-Explorer-Optional-amd64 -Online

Path      :
Online    : True
RestartNeeded : True
```

Figura 7.6: Desactivación de la característica de Windows 10 de Internet Explorer

Al realizar la desactivación de una característica de Windows, al finalizar solicita que reiniciemos el equipo.

7.1.2.2. Aplicaciones de Windows Store

A partir de Windows 8, Microsoft ha implementado una nueva categoría de aplicaciones que comúnmente se les llama apps. Con Windows 10, aparece un repositorio de aplicaciones que se llamada Windows Store.

Para el uso de prácticas docentes no se necesitan estas aplicaciones que vienen predefinidas y resulta conveniente desinstalarlas. Este proceso podría parecer laborioso pero con PowerShell se ha simplificado enormemente.

```
Get-AppxPackage -AllUsers | Remove-AppxPackage
```

Mediante esta simple instrucción se eliminan todas las aplicaciones para todos los usuarios de Windows 10 mediante PowerShell. Si se quisiera realizar la misma operación sobre un equipo externo se ha de emplear el uso de la siguiente instrucción en PowerShell:

```
Invoke-Command -ComputerName 'Equipo'
-ScriptBlock { Get-AppxPackage -AllUsers | Remove-AppxPackage }
```

Comentar que no solo se puede indicar el nombre de un único equipo sino de una colección de ellos. De esta manera se puede realizar sobre un conjunto de equipos ilimitado la misma instrucción que se ha realizado para un equipo local.

7.1.2.3. Los procesos del sistema

La gestión de procesos en un puesto de trabajo es una tarea importante. Se trata de conocer en todo momento el conjunto de programas que se están ejecutando en Windows y los recursos que consumen.

Para ello desde la consola de PowerShell se ejecuta:

```
Get-Process
```

Al cabo de unos instantes aparecen todos los procesos activos en el equipo donde se ha lanzado la instrucción. En la tabla 7.2 se muestra el contenido de las distintas columnas devueltas por el comando **Get-Process**.

En la tabla 7.2 se muestra los cuatro modos de arranque que puede tener un servicio Windows.

Información	Explicación
Handles	Nº descriptores abiertos por el proceso
NPM (K)	Memoria no paginada utilizada por el proceso (tamaño en kilobytes)
PM (K)	Memoria paginada utilizada por el proceso (tamaño en kilobytes)
WS (K)	Rango de trabajo del proceso (kilobytes)
CPU (s)	Tiempo de procesador utilizado por el proceso en el conjunto de procesadores (segundos)
Id	Número identificador proceso (PID)
SI	Número de SessionID. Se usa para identificar al propietario del proceso
ProcessName	Nombre del proceso

Tabla 7.2: Propiedades que muestra el cmdlet Get-Process

Con toda esta información, el administrador puede realizar una consulta en PowerShell para saber los procesos que está lanzando un usuario en una máquina. En la figura 7.7 se muestra un ejemplo:

```
PS D:\> Invoke-Command -ComputerName intelaím4 -ScriptBlock { query session}
NOMBRE DE SESIÒN  NOMBRE DE USUARIO    ID  ESTADO  TIPO  DISPOSITIVO
>services
console          lugio3                1   Conn
rdp-tcp#1        lugio3                3   Activo  rdpwd
rdp-tcp          65536                 Escuchar

PS D:\> Invoke-Command -ComputerName intelaím4 -ScriptBlock { Get-Process | Where-Object {$_.SessionId -eq 3} }
Handles  NPM(K)  PM(K)  WS(K)  CPU(s)  Id  SI  ProcessName  PSComputerName
-----
29        4      3896   6068   0.00    1760  conhost      intelaím4
343       13     6756   13100  7.50    3236  csrss        intelaím4
99        7       4228   7288   0.00    4328  dwm          intelaím4
714       43     41568  76892  79.72   4440  explorer     intelaím4
1822     85    1004972  931160  320.10  4340  MATLAB       intelaím4
309       19     12924  21880   1.98    4608  msseces      intelaím4
217       8       4692   8944   0.03    4108  rdpclip      intelaím4
243       55     10796  14120   0.06    6252  smpd         intelaím4
226       14     9620   11736   0.20    1724  taskhost     intelaím4
208       15     7312   16916   9.393.44  5076  taskmgr      intelaím4
153       8       2112   6136   0.16    4668  VCDDaemon    intelaím4
144       8       7748   11608   0.12    3168  winlogon     intelaím4
```

Figura 7.7: Consulta procesos que ha ejecutado un usuario sobre un equipo remoto

Para detener un proceso con Windows PowerShell, tenemos el cmdlet **Stop-Process**. Se puede buscar en la ayuda para ver los parámetros que puede emplear este cmdlet. Como ejemplo de funcionamiento, se va a detener el proceso del Notepad.exe mediante la siguiente instrucción:

```
Stop-Process -Name notepad
```

Para arrancar un proceso se emplea el cmdlet **Start-Process**. Por ejemplo, si se quiera abrir el block de notas se tendrá que lanzar:

```
Start-Process notepad
```

7.1.2.4. Los servicios del sistema

Los servicios de Windows son programas que, en función de su modo de arranque, se ejecutan permanentemente, o bien se arrancan solo en caso necesario. Además, se pueden desactivar estos servicios o deshabilitarlos para que no arranquen jamás.

Tipo de inicio	Descripción
Manual	El arranque del servicio está autorizado, y lo hace Windows cuando es necesario
Automático	El servicio arranca automáticamente con el inicio de Windows
Automático (inicio retrasado)	El servicio arranca automáticamente, pero con cierto retraso respecto al inicio de la sesión
Deshabilitado	El servicio no arrancará

Tabla 7.3: Descripción de los modos de arranque de los servicios Windows

La manipulación básica que se puede hacer con los servicios son los siguientes:

- **Get-Service:** recupera el estado de uno o varios servicios.
- **Start-Service:** arranca uno o varios servicios.
- **Restart-Service:** reinicio uno o varios servicios.
- **Stop-Service:** detiene uno o varios servicios.
- **Suspend-Service:** pausa uno o varios servicios que están en ejecución.

Con estos cmdlets, un administrador puede realizar muchas cosas sobre los servicios. Puede realizar una búsqueda de servicios que están en ejecución, los que están deshabilitados, etc.. Además puede iniciar, reiniciar, pausar o parar un servicio. Estas operaciones que estamos realizando con un equipo se pueden realizar con uno o varios remotos muy sencillamente.

Los servicios de Windows no se limitan a simples arranques, detenciones o reinicios. También es posible crear nuevos servicios para, por ejemplo, lanzar un ejecutable durante el arranque de Windows, incluso antes de iniciar una sesión un usuario.

Para la manipulación avanzada de los servicios se disponen de los siguientes cmdlets:

- **New-Service:** crea un nuevo servicio.
- **Set-Service:** edita los parámetros de un servicio.

En la figura 7.8 se muestra un ejemplo de cómo crear un nuevo servicio. La creación del servicio se vuelve visible en la consola de servicios (services.msc). El servicio se ejecutará, por defecto, con la cuenta de sistema local; es decir, que el programa que se ejecuta posee todos los permisos en la máquina sobre la que se ha creado.

```
PS D:\> New-Service -Name "Servicio Prueba" -Description `
"Descripcion del servicio Prueba" -BinaryPathName `
E:\temp\miServicio.exe
```

Figura 7.8: Creación de un nuevo servicio en PowerShell

Para un administrador de sistemas es importante poder modificar un servicio existente de una manera sencilla. Por ejemplo, se puede parar el servicio de Windows Search y deshabilitarlo de una manera muy sencilla ya que consume muchos recursos:

```
Set-Service -Name WSearch -StartupType Disabled -Status Stopped
```

Este mismo comando se puede lanzar a un número indefinido de equipos obteniéndose el mismo resultado que para un único equipo.

A pesar que en PowerShell no se dispone de ningún cmdlet para poder eliminar un servicio, es muy sencillo hacerlo utilizando WMI.

7.1.2.5. El sistema operativo de un equipo

En Windows PowerShell se dispone de unos cuantos cmdlets para poder interactuar con un equipo y el sistema operativo. Así, se tienen:

- **Add-Computer:** permite unir el equipo a un dominio o a un grupo de trabajo. Para agregar un equipo a un dominio, se puede realizar con la siguiente instrucción de PowerShell:

```
$cred = Get-Credential
Add-Computer -DomainName "UPVNET" -credential $cred -restart
```

Lo primero que se necesita es tener unas credenciales para poder añadir el equipo al dominio indicado. A continuación, mediante ADD-COMPUTER, se añade al dominio ese equipo. Se le ha añadido al final de la instrucción el parámetro **-restart** para que reinicie el equipo cuando acabe. Si se necesitara más información sobre las opciones de este cmdlet, habría que recurrir a la ayuda de PowerShell.

- **Remove-Computer:** retira el equipo de su dominio. El cmdlet **Remove-Computer** se utiliza específicamente para sacar a los puestos de trabajo de un dominio. Es obligatorio reiniciar el puesto de trabajo tras ejecutar el cmdlet para aplicar los cambios. Una vez reiniciado el equipo, este se encuentra en un grupo de trabajo que se habrá designado mediante el parámetro **-WorkgroupName** (si no se ha especificado nada, el grupo de trabajo será WORKGROUP). Se muestra un ejemplo para sacar un equipo que está en el dominio a un grupo de trabajo.

```
$cred = Get-Credential
Remove-Computer -WorkgroupName "UPVNET" -UnjoinDomainCredential
$cred -Force -Restart
```

- **Rename-Computer:** renombra el equipo.

En este ejemplo se muestra cómo renombrar un equipo local que no está en un dominio AD:

```
Rename-Computer -NewName "NuevoNombreEquipo" -Restart
```

En este caso se le asigna el nuevo nombre (NuevoNombreEquipo) y se reinicia automáticamente para que se efectúe el cambio de nombre. Si el usuario que ejecuta la instrucción no tiene permisos de administrador del equipo, necesita utilizarse el parámetro **-LocalCredential** especificando una cuenta con permisos administrativos.

Si el equipo que se pretende renombrar está en un dominio de Active Directory, se tendría que realizar lo siguiente:

```
$cred = Get-Credential
Rename-Computer -NewName "NuevoNombreEquipo" -DomainCredential
$cred
```

El equipo se ha de reiniciar para que se apliquen los cambios. Una vez reiniciado, el equipo seguirá estando unido al AD pero con el nuevo nombre.

- **Restart-Computer:** reinicia el equipo.

Se puede reiniciar varios equipos (local o remotos) mediante la siguiente instrucción en PowerShell:

```
Restart-Computer -ComputerName Equipo1, Equipo2, Equipo3, ...
```

Es evidente que para poder realizar esta operación deberemos tener permisos en las máquinas para poder reiniciarlas. Se pueden utilizar parámetros sobre este comando para poder personalizar más lo que se necesite.

- **Stop-Computer:** apaga el equipo.

Se puede apagar un equipo local o remoto mediante la siguiente instrucción en PowerShell:

```
$cred = Get-Credential
Stop-Computer -ComputerName Equipo1 -Credential $cred -Force
```

El uso del parámetro **-Force** permite forzar el apagado del equipo correspondiente. Así, si existen sesiones de usuario abiertas en el equipo, se cerrarán automáticamente (hay que tener la precaución de que el trabajo en que se esté realizando en esos momentos no se va a guardar). Si se omite el parámetro **-Force** y existe al menos una sesión de usuario abierta, la petición no se ejecutará y se obtendrá un mensaje de error

- **Reset-ComputerPassword:** reinicia el canal seguro entre el equipo y su dominio. Se dan ocasiones en las que cuando tenemos un equipo que se ha añadido a un dominio, la relación de confianza entre ambos equipos puede romperse. En estos casos, el reinicio del canal seguro permite solucionar este problema.

```
$cred = Get-Credential
Reset-ComputerPassword -Credential $cred
```

La credencial que se ha de emplear es una cuenta de dominio que tenga permisos para realizar esta operación.

- **Test-ComputerSecureChannel:** comprueba el canal seguro. También permite repararlo en caso necesario. Este cmdlet permite comprobar el canal seguro entre el puesto de trabajo y el dominio en el que está presente. Además mediante el parámetro **-repair** permite la posibilidad de reparar el canal seguro si la comprobación fallara.

7.1.2.6. Protección del sistema

Windows posee una característica nativa de protección del sistema una vez se encuentra habilitada, cuyo objetivo es crear puntos de restauración de manera regular, y también tras un cambio a nivel del sistema operativo: instalación de controladores, actualizaciones, etc. Es posible crear un punto de restauración manualmente a través de la interfaz gráfica de Windows, y también mediante Windows PowerShell.

PowerShell dispone de unos cuantos cmdlets que ofrecen la posibilidad de administrar esta característica. Estos cmdlets son:

- **Checkpoint-Computer:** permite crear un punto de restauración. Para la creación de un punto de restauración en un equipo, se ha de lanzar la instrucción siguiente en PowerShell (se ha lanzar la consola con privilegios elevados):

```
Checkpoint-Computer -Description PuntoRestauracion Agosto19
```

- **Disable-ComputerRestore:** desactiva la característica de protección del sistema.

```
Disable-ComputerRestore -Drive C:
```

Permite la desactivación de la protección del sistema en las unidades especificadas en el comando de PowerShell.

- **Enable-ComputerRestore:** activa la característica de protección del sistema.

```
Enable-ComputerRestore -Drive C:
```

La activación de esta característica de Windows debe hacerse obligatoriamente en la unidad donde está instalado el sistema operativo. Tiene la posibilidad de activar la protección del sistema en otras unidades con objeto de aumentar el tamaño máximo disponible para guardar los puntos de restauración.

- **Get-ComputerRestorePoint:** recupera la lista de puntos de restauración guardados en el puesto de trabajo.

El resultado del comando proporciona los detalles para cada uno de los puntos de restauración guardados en el puesto de trabajo. Si no devuelve ningún resultado es porque todavía no se ha creado ningún punto de restauración. En la figura 7.9 se muestran los puntos de restauración creados (se ha de ejecutar en una consola de PowerShell con privilegios administrativos):

```
PS C:\Windows\system32> Get-ComputerRestorePoint | Select-Object CreationTime,Description,SequenceNumber
CreationTime      Description                               SequenceNumber
-----
20190828221707.992360-000 Punto de control programado              14
20190901010001.285567-000 Windows Update                            15
20190907182217.989378-000 Instalador de Módulos de Windows         16
```

Figura 7.9: Listado de los puntos de restauración creados en un equipo mediante PowerShell

- **Restore-Computer:** permite ejecutar una restauración del sistema. Para restaurar un punto de restauración del sistema, se debe averiguar el número de secuencia del punto de restauración que se desea aplicar. Se ha de lanzar las siguientes instrucciones en una consola de PowerShell con elevación de privilegios:

```
(Get-ComputerRestorePoint).SequenceNumber[-1] | Select-Object
    CreationTime, Description, SequenceNumber
Restore-Computer -RestorePoint IdSecuencia
```

Hay que tener en cuenta que cuando se lanza el comando **Restore-Computer** se cierra la sesión en curso y reinicia el equipo. Conviene que se hayan guardado los datos antes para evitar problemas desagradables.

Llegados a este punto, se ha podido observar la enorme potencia, sencillez y comodidad de realizar la administración sobre equipos locales o remotos realizada con PowerShell. No ha sido necesario levantarse del puesto del administrador para realizar todas estas acciones.

7.1.3. Tareas implementadas

Se ha desglosado las tareas que he implementado en varias categorías que se pasan a comentar.

7.1.3.1. Información del sistema

Un apartado importante que un administrador de sistemas utiliza con frecuencia es el apartado de información del sistema donde se obtiene información de uno o varios equipos. Habitualmente se obtiene mediante aplicaciones gráficas que el sistema operativo te ofrece.

Se ha realizado un *script* implementado con PowerShell en el que se obtiene una serie de información del sistema para uno o varios equipos. Para ello se han obtenido objetos de WMI mediante el comando `Get-WmiObject` que genera una gran cantidad de información. Se ha realizado llamadas a una serie de clases para obtener información del sistema operativo, bios, placa base, procesador, discos y del sistema en general. Se ha seleccionado una serie de datos que se han considerado importantes ya que la información que se puede obtener es mucho mayor.

En el diseño de este *script* se requieren dos parámetros obligatorios: un conjunto de equipos sobre los que se va a actuar y el nombre de un fichero que va a almacenar la información del equipo al que no se pueda conectar.

El código implementado para esta función se puede mostrar en el listado de código 7.2.

```
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Muestra informacion variada de un equipo.

.EXAMPLE
'equipo1', 'equipo2' | Get-InfomacionEquipo -logfile '.\
    LogInfomacionEquipos.txt'
```

```

.EXAMPLE
'localhost' | Get-InformacionEquipo -logfile '.\LogInfomacionEquipos.txt'
.EXAMPLE
Get-Content .\equipos.txt | Get-InformacionEquipo -logfile '.\
LogInfomacionEquipos.txt'
.EXAMPLE
Get-InformacionEquipo -nombreEquipo (Get-Content .\equipos.txt) -logfile '
.\LogInfomacionEquipos.txt'

.PARAMETER NombreEquipo
    Equipo sobre el que se quiere obtener informacion.
.PARAMETER logfile
    logfile fichero donde se almacena informacion en caso de tener errores
.

#>

function Get-InformacionEquipo {
    [CmdletBinding()]
    param(
        [Parameter(Mandatory = $true,
            Position = 1,
            ValueFromPipeline = $true,
            ValueFromPipelinePropertyName = $true)]
        [ValidateNotNullOrEmpty()]
        [string[]]$nombreEquipo,
        [Parameter(Mandatory = $true)]
        [string]$logfile
    )
    BEGIN {
        if ($logfile) {
            #Write-Warning 'Eliminando el fichero anterior...'
            remove-item $logfile -ea SilentlyContinue
        }
    }
    PROCESS {
        foreach ($equipo in $nombreEquipo) {
            try {
                #Write-Warning "Obteniendo informacion del equipo $equipo"
                $sesion = New-CimSession -ComputerName $equipo -ErrorAction
                    Stop
                $so = Get-CimInstance -CimSession $sesion -ClassName
                    Win32_OperatingSystem # Sistema operativo
                $bios = Get-CimInstance -CimSession $sesion -ClassName
                    Win32_BIOS #BIOS
                $cs = Get-CimInstance -CimSession $sesion -ClassName
                    Win32_ComputerSystem # Equipo
                $mb = Get-CimInstance -CimSession $sesion -ClassName
                    Win32_BaseBoard # placa base
                $cpu = Get-CimInstance -CimSession $sesion -ClassName
                    Win32_Processor # procesador
                $disco = Get-CimInstance -CimSession $sesion -ClassName
                    Win32_LogicalDisk -Filter "DeviceID='C:'"

                $propiedades = @{NombreEquipo = $equipo
                    Estado = 'Conectado'
                    VersionSP = $so.servicepackmajorversion
                    SistemaOperativo = $so.Caption
                    Arquitectura = $so.OSArchitecture
                    Instalado = $so.ConvertToDateTime($so.
                        InstallDate)
                    SerialBIOS = $bios.SerialNumber
                }
            }
        }
    }
}

```

```

        ModeloPB                = $mb.Product
        Modelo                  = $cs.Modelo
        Fabricante              = $cs.Manufacturer
        Procesador              = $cpu.name
        "Memoria RAM (GB)"      = ([math]::round($cs.TotalPhysicalMemory / 1GB, 2))
        "EspacioLibre (GB)"    = ([math]::round($disco.freespace / 1GB, 2))
        "% disco libre"        = ([math]::round($disco.freespace / $disco.size * 100, 2))
    }
}
catch {
    #Write-Verbose "No puedo conectar con $equipo"
    $propiedades = @{NombreEquipo = $equipo
        Estado                = 'Desconectado'
        VersionSP              = $null
        SistemaOperativo       = $null
        Arquitectura           = $null
        Instalado              = $null
        SerialBios             = $null
        ModeloPB               = $null
        Modelo                  = $null
        Fabricante              = $null
        Procesador              = $null
        "Memoria RAM (GB)"     = $null
        "EspacioLibre (GB)"    = $null
        "% disco libre"        = $null
    }
    if ($logfile) {
        #Write-Warning "Guardando en el fichero log"
        $equipo | Out-File $logfile -Append
    }
}
finally {
    $obj = New-Object -TypeName psubject -Property $propiedades
    Write-Output $obj
}
}
}
END { }
}

```

Listado código 7.2: Get-InformacionEquipo.ps1

Cabe mencionar que se ha comentado los avisos del código para obtener un resultado más limpio. Estos mensajes son interesantes mientras se realizan las pruebas iniciales. Una vez verificado que todo es correcto se recomienda comentar los mensajes de advertencia.

Para mayor comodidad a la hora de lanzar el código se ha importado como un módulo mediante el cmdlet "Import-Module" tal como se observa en el figura 7.10.

```

PS D:\feolde\6_TFG\1_Scripts\Creaciones> Import-Module .\Get-InformacionEquipo.ps1
PS D:\feolde\6_TFG\1_Scripts\Creaciones> |

```

Figura 7.10: Import-Module Get-InformacionEquipo.ps1

ya depende de nosotros: se puede sacar en formato lista, tabla y también exportar a un fichero.

7.1.3.2. Informe de un equipo en HTML

En este apartado se va a realizar mediante *scripts* en PowerShell la obtención de información de uno o varios equipos obteniendo el resultado del mismo en formato HTML para poder ser utilizado desde un navegador web. Es un tanto más sofisticado que el código que se mostró en el apartado 7.1.3.1.

Como paso previo he creado una serie de archivos en PowerShell para obtener diferente información del sistema. El hecho de realizarlo así es que se puede aprovechar dichos archivos en la creación de nuevas funcionalidades que se puedan implementar en un futuro (programación modular). Es por todo sabido de la importancia de la reutilización de funciones ya implementadas y por ello no me voy a extender en este concepto. En definitiva, se han definido una serie de funciones independientes que recopilan cierta información del sistema (para un equipo determinado) y posteriormente mediante el uso de dichas funciones se ha creado un archivo en el que se genera un informe en HTML. En este caso en concreto no se ha encapsulado dichas funciones en un módulo. El hecho de crear un módulo facilita el uso de dichas funciones en otros lugares pero por el momento no se ha realizado.

Se han realizado seis funciones (cada una de ellas en un archivo independiente) que pasa a mostrar:

- Get-InfoSO.ps1.

```
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Muestra informacion del sistema operativo.

.EXAMPLE
    Get-InfoSO -NombreEquipo 'equipo'

.PARAMETER NombreEquipo
    Equipo sobre el que se quiere obtener informacion.
#>

function Get-InfoSO {
    [CmdLetBinding()]

    param (
        [Parameter(Mandatory = $true)]
        [string]$NombreEquipo
    )
    $so = Get-WmiObject -Class Win32_OperatingSystem -ComputerName
        $NombreEquipo
    $propiedades = @{ 'Version SO' = $so.Version;
                    'ServicePack' = $so.ServicePackMajorVersion;
                    'OSBuild' = $so.BuildNumber;
                    'SO' = $so.Caption }
```

```

    New-Object -TypeName psubject -Property $propiedades
}

```

Listado código 7.3: Código Get-InfoSO.ps1

■ Get-InfoSistema.ps1.

```

<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Muestra informacion del sistema.

.EXAMPLE
    Get-InfoSistema -NombreEquipo 'equipo'

.PARAMETER NombreEquipo
    Equipo sobre el que se quiere obtener informacion.
#>

function Get-InfoSistema {
    [CmdLetBinding()]

    param (
        [Parameter(Mandatory = $true)]
        [string]$NombreEquipo
    )
    $cs = Get-WmiObject -Class Win32_ComputerSystem -ComputerName
        $NombreEquipo
    $memo = Get-WmiObject win32_operatingSystem -ComputerName
        $NombreEquipo | Select -property TotalVisibleMemorySize ,
        FreePhysicalMemory
    $propiedades = @{ 'Fabricante' = $cs.Manufacturer;
        'Modelo' = $cs.Model;
        'RAM (GB)' = "{0:N2}" -f ($cs.TotalPhysicalMemory
            / 1GB);
        'RAM (GB) libre' = "{0:N2}" -f ($memo.
            FreePhysicalMemory / 1MB);
        'Cores' = $cs.NumberOfLogicalProcessors;
        'Sockets' = $cs.NumberOfProcessors }
    New-Object -TypeName psubject -Property $propiedades
}

```

Listado código 7.4: Código Get-InfoSistema.ps1

■ Get-InfoDisco.ps1.

```

<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Muestra informacion de los discos de un equipo.

```

```

.EXAMPLE
Get-InfoDisco -NombreEquipo 'equipo'

.PARAMETER NombreEquipo
    Equipo sobre el que se quiere obtener informacion.
#>
function Get-InfoDisco {
    [CmdLetBinding()]

    param (
        [Parameter(Mandatory = $true)]
        [string]$NombreEquipo
    )
    $unidades = Get-WmiObject -Class Win32_LogicalDisk -ComputerName
        $NombreEquipo '
        -Filter "DriveType=3"
    foreach ($unidad in $unidades) {
        $propiedades = @{ 'Unidad' = $unidad.DeviceID;
            'Tamanyo' = $unidad.size / 1GB -as [int];
            #'Libre' = "{0:N2}" -f ($unidad.freespace / 1
                GB);
            'Libre' = $unidad.freespace / 1GB -as [int];
            'Porcentaje' = $unidad.freespace / $unidad.
                size * 100 -as [int]
        }
        New-Object -TypeName psobject -Property $propiedades
    }
}

```

Listado código 7.5: Código Get-InfoDisco.ps1

■ Get-InfoProceso.ps1.

```

<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRPTION
    Muestra informacion de procesos de un equipo.

.EXAMPLE
Get-InfoProceso -NombreEquipo 'equipo'

.PARAMETER NombreEquipo
    Equipo sobre el que se quiere obtener informacion.
#>

function Get-InfoProceso {
    [CmdLetBinding()]

    param (
        [Parameter(Mandatory = $true)]
        [string]$NombreEquipo
    )
    $procesos = Get-WmiObject -Class Win32_Process -ComputerName
        $NombreEquipo
    foreach ($proc in $procesos) {
        $propiedades = @{ 'Nombre proceso' = $proc.name;

```



```

        'Ejecutable' = $proc.ExecutablePath }
    New-Object -TypeName psubject -Property $propiedades
}
}

```

Listado código 7.6: Código Get-InfoProceso.ps1

- Get-InfoServicioRevision.ps1.

```

<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Muestra informacion de los servicios de un equipo.

.EXAMPLE
    Get-InfoServicioRevision -NombreEquipo 'equipo'

.PARAMETER NombreEquipo
    Equipo sobre el que se quiere obtener informacion.
#>

function Get-InfoServicioRevision {
    [CmdLetBinding()]

    param (
        [Parameter(Mandatory = $true)]
        [string]$NombreEquipo
    )
    $srvcs = Get-WmiObject -Class Win32_Service -ComputerName
        $NombreEquipo '
        -Filter "StartMode='Auto' AND State <>'Running'"
    foreach ($srv in $srvcs) {
        $propiedades = @{ 'Nombre Servicio' = $srv.name;
            'Cuenta Logon' = $srv.startname;
            'Nombre Mostrado' = $srv.displayname }
        New-Object -TypeName psubject -Property $propiedades
    }
}

```

Listado código 7.7: Código Get-InfoServicioRevision.ps1

- Get-InfoRed.ps1.

```

<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Muestra informacion de red de un equipo.

.EXAMPLE
    Get-InfoRed -NombreEquipo 'equipo'

```

```

.PARAMETER NombreEquipo
    Equipo sobre el que se quiere obtener informacion.
#>
function Get-InfoRed {
    [CmdLetBinding()]

    param (
        [Parameter(Mandatory = $true)]
        [string]$NombreEquipo
    )
    $red = Get-WmiObject -Class Win32_NetworkAdapter -ComputerName
        $NombreEquipo -Filter "PhysicalAdapter=True"
    foreach ($tarj in $red) {
        $propiedades = @{ 'Nombre NIC' = $tarj.servicename;
            'Velocidad (Mb)' = $tarj.speed / 1Mb -as [int]
            };
            'Fabricante' = $tarj.manufacturer;
            'Direccion MAC' = $tarj.macaddress}
        New-Object -TypeName psubject -Property $propiedades
    }
}

```

Listado código 7.8: Código Get-InfoRed.ps1

En todas las funciones que se han mostrado anteriormente en los Listados de código 7.3, 7.4, 7.5, 7.6, 7.7 y 7.8 se han obtenido unos nombres de propiedades más adecuados que los obtenidos directamente de Get-WmiObject. En el listado de código 7.4 se usa el operador de formato -f con las propiedades de la memoria RAM de modo que se obtiene un valor en gigabytes con dos decimales. Los valores que se obtienen están en bytes (memoria total física) y kilobytes (memoria libre) respectivamente y hay que convertirlas a gigabytes. En el listado de código 7.7 se obtienen varios objetos de WMI y se ha de enumerar utilizando la construcción ForEach. Por último, mencionar que en el listado de código 7.8 se ha tenido que realizar la conversión de la velocidad de bytes a megabytes obteniendo un valor entero.

Una vez comentado brevemente las distintas funciones que se han implementado, se pasa a mostrar el código necesario para construir el HTML.

```

#requires -module EnhancedHTML2
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Genera un informe en formato HIML de uno o varios equipos.
    Por cada equipo especificado se va a generar un archivo HIML.
    Se ha de especificar la ruta donde se van a guardar los archivos.

    Comentar que si ya existieran los archivos se sobrescribieran.

.EXAMPLE
    .\New-InformeHTML -NombreEquipo 'equipo1', 'equipo2' -ruta 'e:\Informe\'

.PARAMETER NombreEquipo
    Equipo o equipos sobre los que se va a generar un archivo HIML.

```

```

.PARAMETER ruta
    Directorio donde se van a escribir los archivos.
#>

[CmdLetBinding()]

param (
    [Parameter(Mandatory = $true,
        ValueFromPipeline = $true,
        ValueFromPipelineByPropertyName = $true)]
    [string[]]$NombreEquipo,

    [Parameter(Mandatory = $true)]
    [string]$ruta
)
BEGIN {
    # importación de las funciones necesarias
    Import-Module .\Get-InfoSO.ps1
    Import-Module .\Get-InfoSistema.ps1
    Import-Module .\Get-InfoDisco.ps1
    Import-Module .\Get-InfoProceso.ps1
    Import-Module .\Get-InfoServicioRevision.ps1
    Import-Module .\Get-InfoRed.ps1
}

PROCESS {

    $estilo = @"
<style>
body {
    color:#333333;
    font-family:Calibri,Tahoma;
    font-size: 10pt;
}

h1 {
    text-align:center;
}

h2 {
    border-top:1px solid #666666;
}

th {
    font-weight:bold;
    color:#eeeeee;
    background-color:#333333;
    cursor:pointer;
}

.odd { background-color:#ffffff; }

.even { background-color:#dddddd; }

.paginate_enabled_next, .paginate_enabled_previous {
    cursor:pointer;
    border:1px solid #222222;
    background-color:#dddddd;
    padding:2px;
    margin:4px;
    border-radius:2px;

```

```

    }

    .paginate_disabled_previous , .paginate_disabled_next {
        color:#666666;
        cursor:pointer;
        background-color:#dddddd;
        padding:2px;
        margin:4px;
        border-radius:2px;
    }

    .dataTables_info { margin-bottom:4px; }

    .sectionheader { cursor:pointer; }

    .sectionheader:hover { color:red; }

    .grid { width:100% }

    .red {
        color:red;
        font-weight:bold;
    }

    .yellow {
        color:yellow;
        font-weight:bold;
    }

    .green {
        color:green;
        font-weight:bold;
    }
}
</style>
"@

foreach($equipo in $nombreequipo) {
    try {
        $Funciona_bien = $true;
        Write-verbose "Comprobando conectividad con $equipo"
        Get-WmiObject -class win32_BIOS -computerName $equipo -EA
            Stop | Out-Null
    } catch {
        Write-Warning "$equipo desconectado"
        $Funciona_bien = $false
    }

    if ($Funciona_bien) {
        $ficheroGuarda = Join-Path -Path $ruta -ChildPath "$equipo.
            html"
        #Sistema operativo
        $parametros = @{ 'AS' = 'List';
            'PreContent' = '<h2>Sistema Operativo</h2>'
        }

        $html_os = Get-InfoSO -NombreEquipo $equipo | ConvertTo-
            EnhancedHTMLFragment @parametros

        #Sistema
        $parametros = @{ 'As' = 'List';

```

```

        'PreContent' = '<h2>Sistema del equipo</h2>'
    }
$html_Sistema = Get-InfoSistema -NombreEquipo $equipo |
    ConvertTo-EnhancedHTMLFragment @parametros

#discos
$parametros = @{ 'As' = 'Table';
    'PreContent' = '<h2>&diam; Discos locales</h2>';
    'EvenRowCssClass' = 'even';
    'OddRowCssClass' = 'odd';
    'MakeTableDynamic' = $true;
    'TableCssClass' = 'grid';
    'Properties' = 'Unidad',
    @{n='Tamanyo (GB)';e={$_.Tamanyo}},
    @{n='Libre (GB)';e={$_.Libre};css={if ($_.
        Libre -lt 20) { 'red' } '
        elseif ($_.Libre -lt 40) { 'yellow' } '
        else { 'green' }}}},
    @{n='% Libre';e={$_.Porcentaje};css={if (
        $_.Porcentaje -lt 20) { 'red' } '
        elseif ($_.Porcentaje -lt 40) { 'yellow'
        } '
        else { 'green' }}}}}
$html_Discos = Get-InfoDisco -NombreEquipo $equipo |
    ConvertTo-EnhancedHTMLFragment @parametros

#procesos
$parametros = @{ 'As' = 'Table';
    'PreContent' = '<h2>&diam; Procesos</h2>';
    'MakeTableDynamic' = $true;
    'TableCssClass' = 'grid' }
$html_Procesos = Get-InfoProceso -NombreEquipo $equipo |
    ConvertTo-EnhancedHTMLFragment @parametros

#Servicio Malo
$parametros = @{ 'As' = 'Table';
    'PreContent' = '<h2>&diam; Servicios a
        chequear</h2>';
    'EvenRowCssClass' = 'even';
    'OddRowCssClass' = 'odd';
    'MakeHiddenSection' = $true;
    'TableCssClass' = 'grid' }

$html_Servicio = Get-InfoServicioRevision -NombreEquipo
    $equipo | ConvertTo-EnhancedHTMLFragment @parametros

#Tarjeta de red
$parametros = @{ 'As' = 'Table';
    'PreContent' = '<h2>&diam; Tarjetas de red</h2>';
    'EvenRowCssClass' = 'even';
    'OddRowCssClass' = 'odd';
    'MakeHiddenSection' = $true;
    'TableCssClass' = 'grid' }

$html_red = Get-InfoRed -NombreEquipo $equipo | ConvertTo-
    EnhancedHTMLFragment @parametros

# se finaliza la configuraci n HIML
$parametros = @{ 'CssStyleSheet' = $estilo;
    'Title' = "Informe del sistema para $equipo";

```

```

        'PreContent'="

# 


```

Listado código 7.9: Código New-InformeHTML.ps1

En el listado de código 7.9 se han de comentar varias cosas importantes. Lo primero de ello es que se necesita el módulo EnhancedHTML2. Para ello se debe lanzar una versión de Powershell como administrador y poner el siguiente código:

```
Install-Module EnhancedHTML2
```

Se emplean dos parámetros obligatorios (uno donde se especifican uno o varios nombres de equipos sobre los que se pretende obtener el informe y otra la ruta de acceso a una carpeta donde se van a almacenar los informes en formato HTML). En el bloque BEGIN se importan las funciones que se van a necesitar para generar el informe. En el bloque PROCESS se indica una hoja de estilos en cascada o CSS para mejorar la apariencia de la hoja en HTML. Como se pueden indicar varios equipos es necesario utilizar la estructura ForEach. Lo primero que realizo es si se puede acceder al equipo ya que en caso contrario no se realiza nada. Se va realizando una llamada a las distintas funciones que se han importado y se muestran en formato fila o en formato tabla pasando el resultado a convertirlo mediante ConvertTo-EnhancedHTMLFragment a una gran cadena HTML que se almacena en una variable. Conviene comentar que a la hora de obtener información de los discos físicos del equipo se han puesto colores (rojo, amarillo y verde) el valor tanto en el espacio libre en gigabytes o en el porcentaje de espacio que hay libre en función de si es menor de 20 (rojo), 40 (amarillo) o el resto de valores (verde). Por último y para generar el HTML final se emplea dos archivos JavaScript (jquery y jquerydatatable) necesarios que se utilizan de la red de distribución de contenido (CDN) basado en la web de Microsoft. Se genera en la ruta que se ha pasado como parámetro un archivo en formato HTML por cada equipo sobre el que se quiere obtener información. En caso de actuar sobre un mismo equipo, dicho archivo se sobrescribe. Dichos archivos se pueden abrir con un navegador web obteniéndose el siguiente resultado:

Informe del sistema para masteraim-01 con fecha: 08/28/2019 23:33:24

Sistema Operativo
 OSBuild : 17134
 ServicePack : 0
 SO : Microsoft Windows 10 Enterprise
 Version SO : 10.0.17134

Sistema del equipo
 Cores : 4
 Fabricante : ASUS
 Modelo : All Series
 RAM (GB) : 7.94
 RAM (GB) libre : 6.39
 Sockets : 1

◆ Discos locales
 Show 10 entries
 Search:

Unidad	Tamaño (GB)	Libre (GB)	% Libre
C:	195	30	15
U:	736	736	100

Showing 1 to 2 of 2 entries
 Previous Next

◆ Procesos
 Show 10 entries
 Search:

Ejecutable	Nombre proceso
	System Idle Process
	System
	Registry
	smss.exe
	csrss.exe
	wininit.exe
	csrss.exe
	services.exe
	Memory Compression
	SecurityHealthService.exe

Showing 1 to 10 of 127 entries
 Previous Next

◆ Servicios a chequear ←

◆ Tarjetas de red ←

Figura 7.14: Resultado HTML obtenido al ejecutar New-InformeHTML.ps1

En la figura 7.14 se puede observar en la parte inferior del informe (indicado mediante flechas rojas) que se puede desplegar. Se trata de una tabla que se ha ocultado pero presionando sobre sus cabeceras se muestra su contenido (tal como se puede ver en la figura 7.15). Es interesante que por defecto las tablas muestran 10 filas pero se pueden hacer más grandes y también se ofrece un cuadro de búsqueda incorporado. Además si se hace “click” sobre los encabezados de columna se ordenan los valores.

◆ Servicios a chequear

Cuenta Logon	Nombre Mostrado	Nombre Servicio
LocalSystem	Cliente de directiva de grupo	gpsvc
NT AUTHORITY\NetworkService	Administrador de mapas descargados	MapsBroker
NT AUTHORITY\NetworkService	Protección de software	sppsvc

◆ Tarjetas de red

Direccion MAC	Fabricante	Nombre NIC	Velocidad (Mb)
E0:3F:49:56:2B:44	Realtek	rt640x64	93

Figura 7.15: Visualización valores Servicios a chequear y tarjetas de red desplegados

Con respecto a las pruebas que se han realizado para esta sección 7.1.3.2 se ha realizado con uno y varios equipos. Como se ha indicado en el listado de código 7.9, se guardan los ficheros HTML generados en la ruta indicada (uno por equipo). En el caso usar un equipo que está apagado o no tiene conexión a dicho equipo aparece un mensaje de advertencia tal como se puede observar en la figura 7.16.

```
PS D:\feolde\6_TFG\1_Scripts\Creaciones> .\New-InformeHTML.ps1 -NombreEquipo dellaim8, aicmudal -ruta E:\HTML
ADVERTENCIA: dellaim8 desconectado
PS D:\feolde\6_TFG\1_Scripts\Creaciones> |
```

Figura 7.16: Aviso de advertencia informando que el equipo no está disponible

Como una posible mejora de este *script* se podría enviar un correo electrónico al administrador para notificarle que se ha lanzado un nuevo informe con formato HTML sobre una serie de equipos previstos. Como se genera un archivo por equipo es probable que se generen bastantes archivos.

Con el archivo en HTML que se genera por equipo se puede obtener una serie de datos importantes para el administrador de una manera sencilla: velocidad de la tarjeta de red, espacio libre de los discos del equipo, cantidad de memoria RAM libre que se dispone, procesos en ejecución, etc.. Con un rápido vistazo se pueden detectar fallos o aplicar acciones correctivas sobre un equipo determinado. Es muy útil esta serie de informes que se generan que facilitan al administrador de los equipos las tareas que ha de realizar. Observando con detenimiento la figura 7.14 se detecta claramente que se dispone poco espacio en el disco C del sistema (se recomienda tener siempre un 10 % del espacio del disco libre porque el rendimiento del mismo desciende muchísimo, lo que no es nada recomendable). Por tanto, nos indica se tendrá que plantear poner un disco de mayor capacidad. En la figura 7.15 se ha detectado que la tarjeta de red tiene una velocidad de 95 Mb. Esta información es importante a la hora de realizar un clonado del sistema en un aula entera ya que en el caso de hacerlo por multicast (tal como se explicó en la sección 4.10) la velocidad de restauración de una imagen va a depender de muchos factores y uno de ellos es la velocidad de las tarjetas de los equipos. Si disponemos de equipos a gigabit y tenemos algún equipo que funciona a 100 Mb, será este equipo el que limite la velocidad al resto y le costará mucho más tiempo.

7.1.3.3. Obtener un monitor de servidores

Se ha desarrollado un sencillo *script* en el que se va a generar un sencillo archivo en HTML con el nombre del servidor, el % de uso de la CPU, la cantidad de memoria RAM libre que posee el equipo, el % del disco C o del sistema que se dispone y la fecha y hora de cuando se obtiene el informe. Una vez generado el archivo HTML con la información adecuada, se adjunta dicho archivo por correo electrónico al administrador del sistema. Una vez más, se puede obtener de una manera muy sencilla información de un montón de servidores. A pesar de la sencillez de esta información que se ha generado tras lanzar este *script*, es útil para el administrador de sistemas que puede saber el consumo de la CPU, la memoria libre del sistema y el porcentaje de espacio libre en el disco del sistema. Con dicha información, el administrador en un momento determinado podrá planificar una parada programada para realizar un mantenimiento más a fondo sobre el equipo que disponga de pocos recursos.

El código es el siguiente:

```
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Obtiene un fichero HTML con información básica de servidores.

.EXAMPLE
Get-MonitorServidores .\equipos.txt -RutaHTML E:\Web.html

.EXAMPLE
```



```

Get-MonitorServidores -nombreEquipo (Get-Content .\equipos.txt) -RutaHTML E
:\Web.html

.PARAMETER NombreEquipo
    Equipo/s sobre el que se quiere obtener informacion.

.PARAMETER RutaHTML
    Ruta completa donde se va a guardar el HTML
#>

function Get-MonitorServidores {
    [CmdletBinding()]
    param(
        [Parameter(Mandatory = $true,
            Position = 1,
            ValueFromPipeline = $true,
            ValueFromPipelineByPropertyName = $true)]
        [ValidateNotNullOrEmpty()]
        [string[]]$nombreEquipo,

        [Parameter(Mandatory = $true)]
        [string]$RutaHTML
    )

    BEGIN {
        # No es necesario pero por si acaso se eliminar todos los Jobs
        Remove-Job -name * -Force
        Import-Module .\Get-MonitorServidores.ps1
    }

    PROCESS{

        #Creaci n HTML
        $Outputreport = "<!DOCTYPE html>
<html>
<head>
    <meta http-equiv=""refresh"" content=""8"" >
    <style>
    table {
        border-collapse: collapse;
        width: 100%;
    }

    th, td {
        text-align: left;
        padding: 8px;
    }

    tr:nth-child(even){background-color: #f2f2f2}

    th {
        background-color: #053F71;
        color: white;
    }
    </style>
    </head>
<body>

    <h2>Alertas Servidores Preproduccion</h2>

    <table>

```

```

        <tr>
            <th>Servidor</th>
            <th>% Uso CPU</th>
            <th>Ram Libre GB</th>
            <th>% C: Libre</th>
            <th>Fecha de datos</th>
        </tr>"
$sonConexion = $false # defino una variable para ver si hay
                    conexion con un equipo
# Se lanza Jobs
ForEach ($equipo in $nombreEquipo)
{
    if (Test-Connection $equipo -Quiet -Count 1)
    { # Compruebo que hay conexion con el equipo
        $sonConexion = $true
        Start-Job -name ($equipo+".UsoCPU") -ScriptBlock {Get-
            WmiObject win32_processor -ComputerName $using:equipo
            | Measure-Object -property LoadPercentage -Average |
            foreach {"{0:N2}" -f ($_.Average)} } -ArgumentList
            $equipo
        Start-Job -name ($equipo+".UsoMEMORIA") -ScriptBlock {Get-
            WmiObject win32_operatingSystem -ComputerName $using:
            equipo |
            Select -property TotalVisibleMemorySize ,
            FreePhysicalMemory } -ArgumentList $equipo
        Start-Job -name ($equipo+".UsoDiscoC") -ScriptBlock {Get-
            WmiObject Win32_LogicalDisk -Filter "DeviceID='C:' " -
            ComputerName $using:equipo |
            Select -property freespace , size } -ArgumentList
            $equipo
    }
}

ForEach ($equipo in $nombreEquipo)
{
    if (Test-Connection $equipo -Quiet -Count 1)
    { # Compruebo que hay conexi n con el equipo
        $sonConexion = $true
        ## ESPERAMOS RESULTADO DE JOB CPU
        try { Wait-Job -name ($equipo+".UsoCpu")
            $UsoCPUfinal = (Receive-Job -Name ($equipo+"
            .UsoCpu") -ErrorAction Stop)
        } catch { $UsoCPUfinal = "" }
        # Se elimina el Job
        Remove-Job -name ($equipo+".UsoCpu") -Force

        ## ESPERAMOS RESULTADO DE JOB USO MEMORIA
        try{ Wait-Job -name ($equipo+".UsoMEMORIA")
            $datos = (Receive-Job -Name ($equipo+"
            .UsoMEMORIA
            ") -ErrorAction stop)
            $MemoriaUsadaFinal = "{0:N2}" -f ($datos.
            FreePhysicalMemory / 1MB)
        } catch { $MemoriaUsadaFinal = "" }
        # Se elimina el Job
        Remove-Job -name ($equipo+".UsoMEMORIA") -Force

        ## ESPERAMOS RESULTADO DE JOB USO C
        try{ Wait-Job -name ($equipo+".UsoDiscoC")
            $datosDisco = (Receive-Job -Name ($equipo+"
            .UsoDiscoC") -ErrorAction Stop)
    }
}

```

```

        $DiscoLibreFinal = [math]::Round($datosDisco.
            freespace / $datosDisco.size * 100, 2)
    } catch { $DiscoLibreFinal = "" }
    # Se elimina el Job
    Remove-Job -name ($equipo+".UsosDiscoC") -Force

    #Se finaliza el HTML
    $Outputreport += "<tr>"
    $Outputreport += "<td>$( $equipo)</td>"
    $Outputreport += "<td>$( $UsoCPUfinal)</td>"
    $Outputreport += "<td>$( $MemoriaUsadaFinal)</td>"
    $Outputreport += "<td>$( $DiscoLibreFinal)</td>"
    $Outputreport += "<td>$(Get-Date)</td>"
    $Outputreport += "</tr>"
    }
}

$Outputreport += "</table>"
$Outputreport += "</body>"
$Outputreport += "</html>"
#Se guarda en el archivo
$Outputreport | out-file "$RutaHTML"

#Finalmente y tras acabar se env a un correo con la info
# se verifica si se dispone de un solo equipo
if ($nombreEquipo.Count -eq 1) {
    if ($conConexion -eq $false) {Write-Warning "Solo se dispone de
        un equipo y no se puede establecer conexion"
    } else {
        try {
            Send-MailMessage -From "InformeMonitorServidores@upv.es" -
                To "fede.olmeda@upv.es" '
            -Subject "Resultado de monitorizacion de servidores " -
                Body "$(Get-Date), se ha generado un informe de
                monitorizacion de servidores" '
            -SmtpServer "smtp.upv.es" -Attachments "$RutaHTML" -
                errorAction Stop
        } catch {
            Write-Warning "No ha sido posible enviar el correo con
                el informe de la monitorizacion de los servidores"
        }
    }
} else {
    try {
        Send-MailMessage -From "InformeMonitorServidores@upv.es" -
            To "fede.olmeda@upv.es" '
        -Subject "Resultado de monitorizacion de servidores " -Body
            "$(Get-Date), se ha generado un informe de
            monitorizacion de servidores" '
        -SmtpServer "smtp.upv.es" -Attachments "$RutaHTML" -
            errorAction Stop
    } catch {
        Write-Warning "No ha sido posible enviar el correo con el
            informe de la monitorizacion de los servidores"
    }
}
}

END {
    Remove-Job -name * -Force
}

```

}

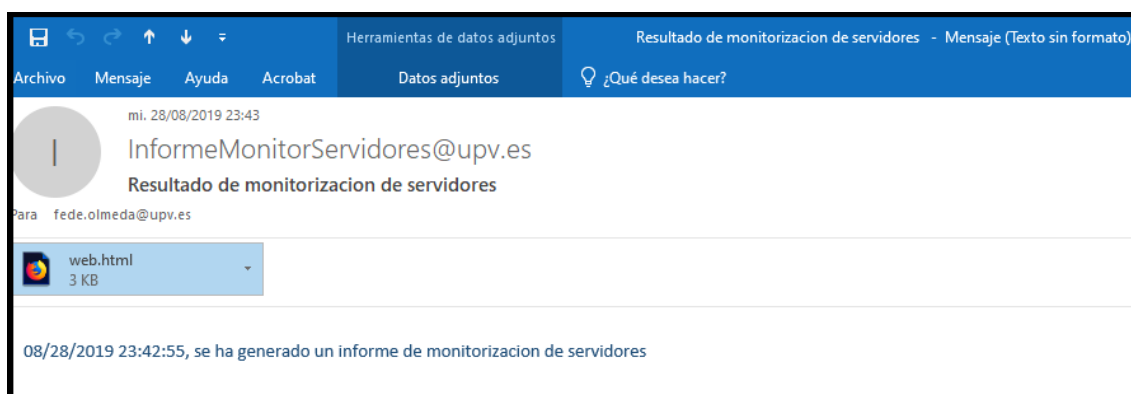
Listado código 7.10: Código Get-MonitorServidores.ps1

PowerShell permite realizar tareas asíncronas mediante el uso de Jobs. En este código se ha utilizado. Se puede observar que como se puede emplear el código para un conjunto de equipos, lo primero que se realiza es verificar que se tiene conexión al equipo (en caso de no conseguir establecer conexión con el equipo no se hace nada). La parte final se muestra cmdLet para enviar un correo electrónico con la información que se ha generado.

En la figura 7.17 se puede observar cómo se ejecuta el *script* desde una consola PowerShell. Se puede observar cómo se van lanzando los diferentes Jobs y están en estado “running” al principio. Se espera a que concluyan y se obtiene los resultados de cada uno de ellos. Al completarse las tareas de todos los Jobs que se han lanzado, se genera el archivo HTML que se envía por correo electrónico. Para ello consulto mi bandeja de entrada y veo que he recibido un correo electrónico cuyo asunto es “Resultado de monitorización de servidores” tal como se puede observar en la figura 7.18. Se puede observar que tiene un adjunto (web.html) que al abrirlo con un navegador web se abre y se muestra tal como se puede observar en la figura 7.19.

```
PS D:\feolde\6_TFG\1_Scripts\Creaciones> Get-MonitorServidores (Get-Content .\equipos.txt) -RutaHTML e:\web.html
```

Id	Name	PSJobTypeName	State	HasMoreData	Location	Command
4538	dellaim1.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject win32_pr...
4540	dellaim1.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject win32_op...
4542	dellaim1.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject Win32_Lo...
4544	dellaim2.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject win32_pr...
4546	dellaim2.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject win32_op...
4548	dellaim2.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject Win32_Lo...
4550	intelaime2.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject win32_pr...
4552	intelaime2.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject win32_op...
4554	intelaime2.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject Win32_Lo...
4556	intelaime3.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject win32_pr...
4558	intelaime3.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject win32_op...
4560	intelaime3.UsocPU	BackgroundJob	Running	True	localhost	Get-WmiObject Win32_Lo...
4538	dellaim1.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject win32_pr...
4540	dellaim1.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject win32_op...
4542	dellaim1.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject Win32_Lo...
4544	dellaim2.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject win32_pr...
4546	dellaim2.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject win32_op...
4548	dellaim2.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject Win32_Lo...
4550	intelaime2.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject win32_pr...
4552	intelaime2.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject win32_op...
4554	intelaime2.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject Win32_Lo...
4556	intelaime3.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject win32_pr...
4558	intelaime3.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject win32_op...
4560	intelaime3.UsocPU	BackgroundJob	Completed	True	localhost	Get-WmiObject Win32_Lo...

Figura 7.17: Ejecución *script* Get-MonitorServidores**Figura 7.18:** Correo electrónico recibido tras ejecutar el *script* Get-MonitorServidores

Alertas Servidores Produccion				
Servidor	% Uso CPU	Ram Libre GB	% C: Libre	Fecha de datos
dellaim1	1.50	28,72	33.96	08/28/2019 23:42:51
dellaim2	1.50	28,75	34.72	08/28/2019 23:42:51
intelaim2	2.00	46,61	56.85	08/28/2019 23:42:51
intelaim3	1.50	45,45	58.19	08/28/2019 23:42:51

Figura 7.19: Archivo HTML generado por el *script* Get-MonitorServidores

En cuanto a las pruebas que se han realizado con este *script*:

- Un solo equipo sin conexión.

```
PS D:\feolde\6_TFG\1_Scripts\Creaciones> Get-MonitorServidores dellaim8 -RutaHTML e:\web.html
ADVERTENCIA: Solo se dispone de un equipo y no se puede establecer conexion
```

Figura 7.20: Prueba ejecución Get-MonitorServidores con un equipo sin conexión

- Un solo equipo con conexión.

```
PS D:\feolde\6_TFG\1_Scripts\Creaciones> Get-MonitorServidores dellaim1 -RutaHTML e:\web.html

Id      Name                PSJobTypeName State      HasMoreData Location      Command
--      -
4562    dellaim1.UsocPU    BackgroundJob  Running    True       localhost    Get-WmiObject win32_pr...
4564    dellaim1.UsocPU    BackgroundJob  Running    True       localhost    Get-WmiObject win32_op...
4566    dellaim1.UsocPU    BackgroundJob  Running    True       localhost    Get-WmiObject win32_Lo...
4562    dellaim1.UsocPU    BackgroundJob  Completed  True       localhost    Get-WmiObject win32_pr...
4564    dellaim1.UsocPU    BackgroundJob  Completed  True       localhost    Get-WmiObject win32_op...
4566    dellaim1.UsocPU    BackgroundJob  Completed  True       localhost    Get-WmiObject win32_Lo...
```

Figura 7.21: Prueba ejecución Get-MonitorServidores con un equipo con conexión
Al acabar se envía un correo electrónico con el resultado.

- Varios equipos y uno de ellos sin conexión.

```
PS D:\feolde\6_TFG\1_Scripts\Creaciones> Get-MonitorServidores dellaim1, dellaim8 -RutaHTML e:\web.html

Id      Name                PSJobTypeName State      HasMoreData Location      Command
--      -
4568    dellaim1.UsocPU    BackgroundJob  Running    True       localhost    Get-WmiObject win32_pr...
4570    dellaim1.UsocPU    BackgroundJob  Running    True       localhost    Get-WmiObject win32_op...
4572    dellaim1.UsocPU    BackgroundJob  Running    True       localhost    Get-WmiObject win32_Lo...
4568    dellaim1.UsocPU    BackgroundJob  Completed  True       localhost    Get-WmiObject win32_pr...
4570    dellaim1.UsocPU    BackgroundJob  Completed  True       localhost    Get-WmiObject win32_op...
4572    dellaim1.UsocPU    BackgroundJob  Completed  True       localhost    Get-WmiObject win32_Lo...
```

Figura 7.22: Prueba ejecución Get-MonitorServidores con varios equipos y uno sin conexión

Como se puede ver en la figura 7.22 se realiza la búsqueda de información en el equipo que dispone conexión y no se realiza nada sobre el equipo sin conexión. Al concluir se envía un correo electrónico con el resultado.

Antes concluir este apartado y con el fin de evitar tener que lanzar manualmente este *script*, se hace imprescindible generar una tarea programada que se encargue de lanzar automáticamente el *script* que se puede ver en el listado de código 7.10.

Es importante comentar que el usuario que utilizo del dominio es un usuario sin privilegios en el equipo desde donde trabajo (es una práctica recomendada y necesaria para mejorar la seguridad del equipo). Para lanzar una tarea programada con un usuario con privilegios reducidos es necesario añadirlo para poder iniciar sesión como proceso

por lotes. La información se puede consultar en [50]. Para ello, lo primero es ejecutar como administrador desde una consola la instrucción `gpedit.msc` que abre el editor de directivas de grupo local. Hay que ir a **Configuración del equipo** → **Configuración de Windows** → **Configuración de seguridad** → **Directivas locales** → **Asignación de derechos de usuario**. Se debe añadir el usuario que se desee a la entrada **Iniciar sesión como proceso por lotes** tal como se observa en la figura 7.23.

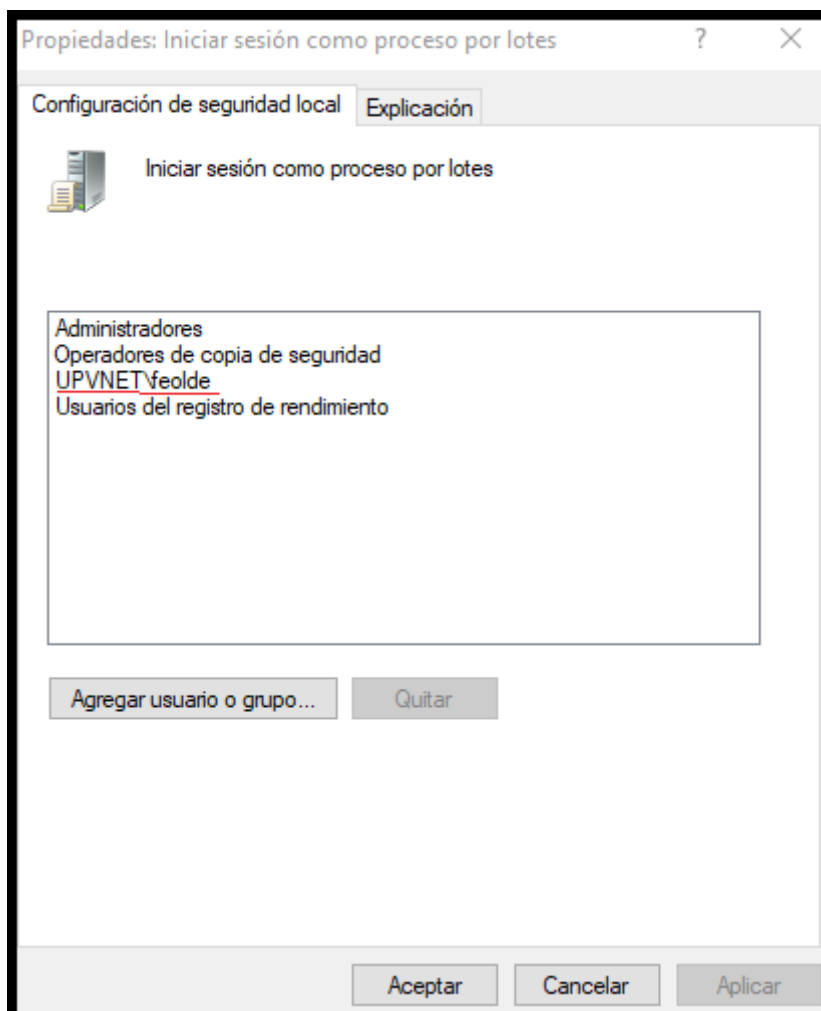


Figura 7.23: Añadir usuario a las propiedades de inicio sesión como proceso por lotes

Una vez añadido el usuario que se quiere utilizar para lanzar la tarea programada, es necesario que se apliquen los cambios para ello y desde una consola como administrador se ha de lanzar un comando con el fin de que se fuerce a que se actualicen las directivas. Se puede ver en la figura 7.24.

```
Administrador: Windows PowerShell
PS C:\Windows\system32> gpupdate /force
Actualizando directiva...

La actualización de la directiva de equipo se completó correctamente.
Se completó correctamente la Actualización de directiva de usuario.

PS C:\Windows\system32>
```

Figura 7.24: Forzado de actualización de las directivas del equipo

Se ha creado un pequeño *script* que realiza la llamada a Get-MonitorServidores, tal como se puede ver en el listado de código 7.11.

```
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Llamada a la ejecución de Get-MonitorServidores
    para lanzarlo en una tarea programada
#>
Import-Module .\Get-MonitorServidores.ps1
Get-MonitorServidores -nombreEquipo (Get-Content e:\Servidores2.txt) -
RutaHTML E:\Web.html
```

Listado código 7.11: Código TareaMonitorServidores.ps1

A continuación se ha creado otro *script* para crear la tarea programa. El código está en el listado de código 7.12.

```
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Se crea una tarea programada para lanzar un script en PowerShell.
    Se pide la contraseña del usuario
#>

$script1 = "D:\feolde\6_TFG\1_Scripts\Creaciones\TareaMonitorServidores.ps1"
$usuario = "UPVNET\feolde"
$SecurePassword = Read-Host "Contraseña de $usuario" -AsSecureString
$Credenciales = New-Object System.Management.Automation.PSCredential -
ArgumentList $usuario, $SecurePassword
$Password = $Credenciales.GetNetworkCredential().Password
$Accion = New-ScheduledTaskAction '
-Execute PowerShell '
-Argument "-File $script1" '
-WorkingDirectory "D:\feolde\6_TFG\1_Scripts\Creaciones\"
$Trigger = New-ScheduledTaskTrigger -Daily -At 8:30am
$opciones = New-ScheduledTaskSettingsSet
Register-ScheduledTask '
-Action $Accion '
-Trigger $Trigger '
-TaskName "Lanzar monitorServicios" '
-Description "Tarea programada generada por la consola Get-
MonitorServidores" '
-user $usuario '
-Password $Password
```

Listado código 7.12: Código LanzarTareaProgramadaMonitorServidores.ps1

Se ha lanzado como se puede ver en la figura 7.25 y se ha logrado disponer de una tarea programada para que diariamente genere un informe de los servidores del AIM que se enviará por correo electrónico al administrador del sistema de dicha área.

```
PS D:\feolde\6_TFG\1_Scripts\Creaciones> D:\FEOLDE\6_TFG\1_Scripts\Creaciones\LanzarTareaProgramadaMonitorServidores.ps1
TaskPath          TaskName          State
-----
\                  Lanzar_monitorServicios  Ready
```

Figura 7.25: Generación de una tarea programa para monitorizar los servidores del área

7.1.3.4. Copia de seguridad

Una parte importante que realiza un administrador de sistemas es la realización de copias de los datos de diversos equipos con el fin de poder recuperar la información (toda o una parte importante de ella) en caso de problemas ya sea por fallo físico del disco, borrado accidental, etc..

Se ha creado un sencillo *script* que realiza una copia de un directorio origen en un directorio destino (son parámetros que se han de introducir en el código) utilizando la herramienta de Microsoft Robocopy. En [51] se muestran los parámetros soportados por esta utilidad que se incluye en los sistemas operativos posteriores a Windows 7 (anteriormente se debía descargar la herramienta de Internet). Se ha utilizado la información de [52] para incluir los códigos de error de esta utilidad en el *script* que se ha implementado para esta funcionalidad de las copias de seguridad.

```
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Realiza una copia de seguridad de un directorio completo.
    Se utiliza la utilidad de Microsoft Robocopy.
    Al finalizar se env a un correo electrónico con el log.

.EXAMPLE
    Get-Backup -RutaOrigen 'E:\src' -RutaDestino 'E:\dst'

.PARAMETER RutaOrigen
    Directorio origen de la copia de seguridad.
.PARAMETER RutaDestino
    Directorio destino de la copia de seguridad.

#>

Function Get-Backup {
    [CmdletBinding()]
    param(
        [Parameter(Mandatory = $true,
            Position = 1,
            ValueFromPipeline = $true,
            ValueFromPipelinePropertyName = $true)]
        [ValidateNotNullOrEmpty()]
        [string]$RutaOrigen,
        [Parameter(Mandatory = $true)]
        [string]$RutaDestino
    )

    $DirectorioLog = "E:\Robocopy\Logs"
```



```

$FicheroLog = "E:\Robocopy\Logs\Robocopy.log" # se indica una ruta fija
$EstadoBackup = "Desconocido"
$Asunto = "" # para indicar en el correo electronico que se envia al final
$CodigoError= ""
$MotivoError = ""

# Se comprueba si el directorio origen existe. Si no existe # no se hace nada
if (Test-Path -Path $RutaOrigen)
{
    # Se comprueba si el directorio de los logs existe. Si no existe se crea
    if (!(Test-Path -path $DirectorioLog))
    {
        New-Item $DirectorioLog -type directory
    }

    # Se comprueba si el destino existe. Si no existe se crea
    if (!(Test-Path -path $RutaDestino))
    {
        New-Item $RutaDestino -type directory -ErrorAction SilentlyContinue
    }

    robocopy $RutaOrigen $RutaDestino /MIR /FFT /R:2 /LOG:
        $FicheroLog /XA:S /XD *RECYCLE.BIN*

    # Control de errores obtenidos de Robocopy
    Switch ($LASTEXITCODE)

    0 {
        $CodigoError = "0"
        $MotivoError = "No hay cambios y no se han copiado archivos ."
        $EstadoBackup = "Correcto"
    }
    1 {
        $CodigoError = "1"
        $MotivoError = "Uno o mas archivos se copiaron correctamente."
        $EstadoBackup = "Correcto"
    }
    2 {
        $CodigoError = "2"
        $MotivoError = "Ficheros o directorios adicionales se han detectado. Revise el fichero log ' para mayor informacion."
        $EstadoBackup = "Correcto"
    }
    3 {
        $CodigoError = "3"
        $MotivoError = "Algunos ficheros se han copiado. Ficheros adicionales estaban presentes."
        $EstadoBackup = "Correcto"
    }
    4 {
        $CodigoError = "4"
        $MotivoError = "Se han detectado ficheros o directorios no coincidentes. Revise el fichero log."
    }
}

```

```
    $EstadoBackup = "Error"
}
5 {
    $CodigoError = "5"
    $MotivoError = "Algunos archivos fueron copiados. Algunos
        archivos no coinciden '
        No se encontro ningun fallo."
    $EstadoBackup = "Error"
}
6 {
    $CodigoError = "6"
    $MotivoError = "Existen archivos adicionales y archivos no
        coincidentes. No se copiaron archivos '
        ni se encontraron fallos. Esto significa que los
        archivos ya existen en el directorio de destino."
    $EstadoBackup = "Error"
}
7 {
    $CodigoError = "7"
    $MotivoError = "Se copiaron archivos, hubo una falta de
        coincidencia de archivos y se encontraron archivos
        adicionales."
    $EstadoBackup = "Error"
}
8 {
    $CodigoError = "8"
    $MotivoError = "Algunos archivos o directorios no se
        pudieron copiar y se super el limite de reintentos."
    $EstadoBackup = "Error"
}
9 {
    $CodigoError = "9"
    $MotivoError = "Fallo en la copia."
    $EstadoBackup = "Error"
}
10 {
    $CodigoError = "10"
    $MotivoError = "[FALLO]. Fallo Extra"
    $EstadoBackup = "Error"
}
11 {
    $CodigoError = "11"
    $MotivoError = "[FALLO]. OK Copia + Fallo copia extra."
    $EstadoBackup = "Error"
}
12 {
    $CodigoError = "12"
    $MotivoError = "[Fallo]. Fallo discrepante "
    $EstadoBackup = "Error"
}
13 {
    $CodigoError = "13"
    $MotivoError = "[Fallo]. OK Copia + fallo discrepante copia
        ."
    $EstadoBackup = "Error"
}
14 {
    $CodigoError = "14"
    $MotivoError = "[Fallo]. Fallo discrepante en datos extra"
    $EstadoBackup = "Error"
}
15 {
```

```

        $CodigoError = "15"
        $MotivoError = "[Fallo] OK Copia + Fallo discripante en
            copia datos extra."
        $EstadoBackup = "Error"
    }
    16 {
        $CodigoError = "16"
        $MotivoError = "[*** ERROR FATAL ***] Robocopy no copi
            ning n archivo. Verifique los par metros de '
            la l nea de comando y verifique que Robocopy tenga
            suficientes derechos para escribir en la carpeta de
            destino."
        $EstadoBackup = "Error"
    }
    default {
        $CodigoError = "Desconocido"
        $MotivoError = "Motivo desconocido"
        $EstadoBackup = "Error"
    }
} # Fin del switch

# Para enviar por correo
$Asunto = "[" + $EstadoBackup + "]" + " " + " " + $MotivoError + "
    CodigoError: " + $CodigoError

# Se env a el correo electr nico
try {
    Send-MailMessage -From "CopiaSeguridad@upv.es" -To "fede.
        olmeda@upv.es" '
        -Subject "$Asunto" '
        -Body "$(Get-Date), se ha realizado una copia de
            seguridad de $RutaOrigen en $RutaDestino" '
        -SmtpServer "smtp.upv.es" '
        -Attachments "$FicheroLog" -errorAction Stop
} catch {
    Write-Warning "No ha sido posible enviar el correo con la
        copia de seguridad de $RutaOrigen en $RutaDestino"
}

# Se elimina el log
Remove-item $FicheroLog
} else {
    Write-Warning "No existe $RutaOrigen"
}
}

```

Listado código 7.13: Código Get-Backup.ps1

Tal como se puede observar en el listado de código 7.13, se ha implementado una función que tiene dos parámetros de entrada: el directorio origen y el directorio destino. Para darle mayor robustez al diseño del *script*, lo primero que se comprueba es si existe el directorio origen. Si no existe muestra un mensaje de advertencia indicando que no existe el origen y acaba. Si existe el origen, se verifica si existe el directorio donde se va a albergar el log de la copia. En el diseño que se ha implementado se fija un directorio concreto del disco duro aunque se podría haber utilizado como un parámetro más en la función. En caso de no existir dicho directorio de log se crea. Algo parecido se realiza con el directorio destino: si no existe se crea. A continuación se realiza la llamada a Robocopy

con unos parámetros determinados. Se ha controlado la salida de errores de Robocopy de la información obtenida en [52]. Finalmente se envía por correo electrónico (tal como se puede observar en la figura 7.26) adjuntando el archivo log y se procede al borrado del log del disco.

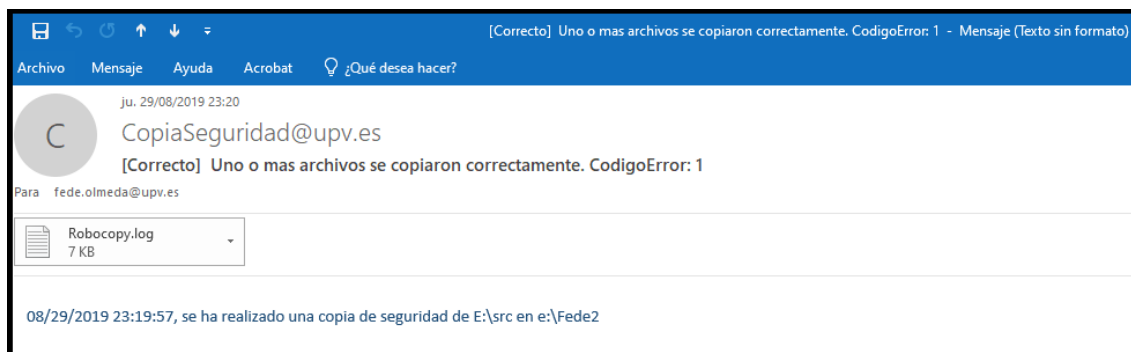


Figura 7.26: Correo electrónico recibido de la función Get-Backup

7.1.3.5. Limpieza de archivos temporales

No nos podemos hacer una idea de los archivos temporales que se generan en un equipo informático con el sistema operativo de Microsoft. Si además tenemos en cuenta que muchos de estos equipos se utilizan para la realización de prácticas docentes y por ellos aparecen multitud de alumnos que inician sesión con sus credenciales del dominio alumno y upvnet (en función de si es trabajador o alumno de la universidad), el espacio ocupado por estos archivos temporales no es para nada despreciable. Aunque se emplean herramientas como Ccleaner ([53]) o similares, es necesario crear un *script* que automatice la limpieza de estos archivos temporales en los equipos.

Se ha implementado un pequeño código en PowerShell que va a resolver de un modo automático la eliminación de los archivos temporales almacenados en el disco del sistema (generalmente disco C). La ubicación donde se almacenan los archivos temporales en Windows son:

- C:\Windows\Temp.
- C:\Windows\Prefetch.
- C:\Documents and Settings*\Local Settings\Temp.
- C:\Users*\Appdata\Local\Temp.

El código se puede ver en el listado de código 7.14. En el diseño de este código, se ha creado una variable con las rutas que se quieren borrar. El único parámetro de la función puede aceptar varios valores y por ello se necesita hacer un recorrido por los mismo mediante un bucle “foreach”. Para cada equipo se comprueba si tiene conexión y en caso de no estar accesible nos da un mensaje de alerta. Se lanza la eliminación de los archivos temporales en aquellos equipos que tienen conexión. Realmente, es un *script* sencillo pero eficaz.

```
<#
.NOTES
```

```

Creado por:      Fede Olmeda
Creado en:       @2019

.DESCRIPTION
  Realiza limpieza de archivos temporales de un equipo con sistema
  operativo Microsoft.

.EXAMPLE
  Clean-ArchivosTemporales -nombreEquipo (Get-Content .\equipos.txt)

.EXAMPLE
  Clean-ArchivosTemporales -nombreEquipo 'equipo'

.PARAMETER NombreEquipo
  Equipo sobre el que se quiere obtener informacion.
#>

Function Clean-ArchivosTemporales {
  [CmdletBinding()]
  param(
    [Parameter(Mandatory = $true,
              Position = 1,
              ValueFromPipeline = $true,
              ValueFromPipelinePropertyName = $true)]
    [ValidateNotNullOrEmpty()]
    [string[]]$nombreEquipo
  )
  Begin{
    Import-Module .\Get-InfoProceso.ps1
  }
  PROCESS {
    # Se define un array con la ruta de archivos temporales
    $DirTemporal = @( "C:\Windows\Temp\*", "C:\Windows\Prefetch\*" '
                    , "C:\Documents and Settings\*\Local Settings\temp\*" '
                    , "C:\Users\*\AppData\Local\Temp\*" )

    foreach ($equipo in $nombreEquipo) {
      $conexion = Test-Connection $equipo -Quiet -Count 1
      If ($conexion -eq 'True'){
        Write-Verbose "Se procede a la limpieza de los archivos
          temporales del equipo $equipo"
        try {
          Invoke-Command -ComputerName $equipo -ScriptBlock {
            Remove-item -Path $using:DirTemporal -force -recurse
          } '
            -ArgumentList $DirTemporal -ErrorAction
              SilentlyContinue
          Write-Verbose "Ha finalizado el proceso de limpieza de los
            archivos temporales del equipo $equipo"
        } catch {
          Write-Warning "Se han detectado problemas en la
            limpieza de los archivos temporales del equipo
              $equipo"
        }
      } else {
        Write-Warning "No se ha podido conectar con el equipo
          $equipo"
      }
    }
  }
}
END {}

```

}

Listado código 7.14: Código Clean-ArchivosTemporales.ps1

En la figura 7.27 se muestra un par de ejecuciones de la función de limpieza de los temporales de un equipo. En la primera llamada aparece una alerta puesto que no puede acceder al equipo "dellaim8". No es necesario el parámetro `-verbose` puesto que se trata de un mensaje de alerta. En el segundo caso, muestra información de lo que hace. Si no se le pone el parámetro `-verbose` finalizaría la función de limpieza y no mostraría nada (este modo de proceder es muy útil cuando se está desarrollando el código de la función que se quiere implementar para resolver una necesidad).

```
PS D:\feolde\6_TFG\1_Scripts\Creaciones> Clean-ArchivosTemporales -nombreEquipo dellaim8 -Verbose
ADVERTENCIA: No se ha podido conectar con el equipo dellaim8

PS D:\feolde\6_TFG\1_Scripts\Creaciones> Clean-ArchivosTemporales -nombreEquipo llaim-12 -Verbose
DETALLADO: Se procede a la limpieza de los archivos temporales del equipo llaim-12
DETALLADO: Ha finalizado el proceso de limpieza de los archivos temporales del equipo llaim-12
```

Figura 7.27: Ejecución de función "Clean-ArchivosTemporales"

7.1.4. Otras tareas sencillas pero útiles y necesarias

7.1.4.1. Conocer la fecha de encendido de un equipo

Mediante la implementación de este sencillo *script*, visualizo la fecha de encendido de un equipo determinado. Es un código muy sencillo pero es muy interesante para el administrador de sistemas. Por ejemplo, se dispone de un reinicio automático en los servidores de cálculo semanalmente a una hora determinada. Con la ejecución de este código puedo verificar de una manera muy rápida que realmente se enciende en la fecha adecuada. Si se detecta encendidos en otras fechas puede alertar al administrador que tendrá que hacer un análisis más profundo del equipo para averiguar el por qué de ese encendido no programado. Se tendrá que tomar las medidas necesarias para esclarecer el problema y dar una solución que garantice el correcto funcionamiento.

El código se muestra en el listado de código 7.15.

```
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Se muestra la fecha en que se enciende un equipo.

.EXAMPLE
    Get-FechaEncendidoEquipo -nombreEquipo 'Equipo'

.PARAMETER NombreEquipo
    Equipo sobre el que se quiere obtener informacion.
#>
Function Get-FechaEncendidoEquipo {
    [CmdletBinding()]
    param(
        [Parameter(Mandatory = $true ,
```

```

        Position = 1,
        ValueFromPipeline = $true,
        ValueFromPipelineByPropertyName = $true)]
    [ValidateNotNullOrEmpty()]
    [string]$nombreEquipo
)
# Compruebo que hay conexión con el equipo
if (Test-Connection $nombreEquipo -Quiet -Count 1) {
    $datos = Invoke-Command -ComputerName $computer -scriptblock {Get-
        EventLog -LogName system -InstanceId 12 | Select TimeGenerated}
    #muestro los datos con el formato de fecha que quiero
    $datos | Select-Object -Property PSComputerName, @{n='Fecha
        reinicio'; e = { Get-Date $_.TimeGenerated -Format "dddd dd/MM/
        yyyy HH:mm" }}
} else {
    Write-Warning "No consigo establecer conexión con el equipo
        $nombreEquipo"
}
}

```

Listado código 7.15: Código Get-FechaEncendidoEquipo.ps1

En la figura 7.28 se muestra el resultado de ejecución del *script* mostrado en el listado de código 7.15. Este equipo tiene programado un reinicio automático todos los lunes a las 13:00 horas. Se puede ver claramente que se ha encendido fuera de su funcionamiento programado. Se ha detectado que han habido problemas con el suministro eléctrico en esas fechas. Se demuestra la importancia de estos pequeños códigos en PowerShell que ayudan al administrador a realizar un seguimiento más profundo de los equipos.

```

PS D:\feolde\6_TFG\1_Scripts\Creaciones> Get-FechaEncendidoEquipo dellaim1
PSComputerName Fecha reinicio
-----
dellaim1        viernes 30/08/2019 18:57
dellaim1        lunes 26/08/2019 13:03
dellaim1        martes 20/08/2019 04:03
dellaim1        martes 20/08/2019 04:03
dellaim1        martes 20/08/2019 04:03
dellaim1        martes 20/08/2019 04:03
dellaim1        martes 20/08/2019 03:58
dellaim1        lunes 19/08/2019 13:51
dellaim1        lunes 12/08/2019 13:04
dellaim1        lunes 05/08/2019 13:04
dellaim1        lunes 05/08/2019 10:02
dellaim1        jueves 01/08/2019 14:34
dellaim1        lunes 29/07/2019 13:04
dellaim1        lunes 22/07/2019 13:04
dellaim1        lunes 15/07/2019 13:11
dellaim1        lunes 15/07/2019 13:11
dellaim1        lunes 15/07/2019 13:11

```

Figura 7.28: Ejecución de función "Get-FechaEncendidoEquipo"

7.1.4.2. Firma de los scripts mediante certificado

PowerShell tiene políticas de ejecución configurables que controlan qué condiciones se requieren para que se ejecute un script o una configuración. Se puede establecer una política de ejecución para múltiples ámbitos: computadora, usuario y proceso actual. Aunque las políticas de ejecución se pueden omitir fácilmente y no están diseñadas para

restringir a los usuarios, sino protegerlos de violar las políticas de forma involuntaria.

Las políticas disponibles son:

Política	Descripción
AllSigned	Solo se puede ejecutar scripts firmados por un editor de confianza.
Bypass	Sin restricciones. Se pueden ejecutar todos los scripts de Windows PowerShell.
Default	Normalmente RemoteSigned, pero se controla a través de ActiveDirectory.
RemoteSigned	Los scripts descargados deben estar firmados por un editor de confianza antes de que puedan ejecutarse.
Restricted	No se pueden ejecutar scripts. Windows PowerShell solo se puede utilizar en modo interactivo.
Undefined	N/A.
Unrestricted	Similar a Bypass

Tabla 7.4: Políticas de ejecución de PowerShell

Se pueden modificar las políticas de ejecución actuales utilizando el cmdlet "Set-ExecutionPolicy". Mediante el cmdlet "Get-ExecutionPolicy -list" se muestra todas las políticas de ejecución para la sesión actual, tal como se puede observar en la figura 7.29.

```
PS E:\> Get-ExecutionPolicy -List

Scope ExecutionPolicy
-----
MachinePolicy Undefined
UserPolicy Undefined
Process Undefined
CurrentUser Undefined
LocalMachine AllSigned
```

Figura 7.29: Listado de políticas de ejecución para la sesión actual

Con la configuración de la política de ejecución de la figura 7.29 se necesita que el *script* esté firmado para poderse ejecutar. En la figura 7.30 se puede observar el contenido del archivo "Fecha2.ps1" (simplemente muestra la fecha actual). Seguidamente, muestra el error al ejecutarlo porque el archivo no está firmado.

```
PS E:\Powershell> Get-Content .\Fecha2.ps1
Get-date

PS E:\Powershell> .\Fecha2.ps1
.\Fecha2.ps1 : No se puede cargar el archivo E:\Powershell\Fecha2.ps1. El archivo
E:\Powershell\Fecha2.ps1 no está firmado digitalmente. No se puede ejecutar este script
en el sistema actual. Para obtener más información acerca de la ejecución de scripts y la
configuración de la directiva de ejecución, consulta about_Execution_Policies en
https://go.microsoft.com/fwlink/?LinkID=135170.
En línea: 1 Carácter: 1
+ .\Fecha2.ps1
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess

PS E:\Powershell>
```

Figura 7.30: Ejecución de un *script* sin firmar

Para poder firmar los *scripts* creados para que no haya problemas de ejecución tanto en local como en remoto, se ha optado por la creación de un certificador autofirmado. El

proceso se muestra en el listado de código 7.16. No hay mucho que explicar porque se ha comentado el código y es bastante claro.

```

# requires -runasadministrator
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Este script genera un certificado autofirmado
    para firmar scripts generados con PowerShell
#>

$NombreCert = "FedeOlmeda"
$RutaExportacion = "E:\Certificados\MiCertificadoFirma.pfx"
$Password = ConvertTo-SecureString -String "UnaC0ntr1S!3c" -Force -
    AsPlainText

#Se crea el certificado
New-SelfSignedCertificate -Type CodeSigningCert -FriendlyName "Certificado
    firma Fede Olmeda" '
    -Subject $NombreCert -CertStoreLocation Cert:\CurrentUser\My '
    | Export-PfxCertificate -FilePath $RutaExportacion -Password $Password

Write-Output "El certificado PFX $NombreCert se ha exportado en
    $RutaExportacion"

# Importar certificado PFX
Import-PfxCertificate -FilePath $RutaExportacion '
    -CertStoreLocation "cert:\LocalMachine\My" -Password $Password

# Importar certificado PFX Root
Import-PfxCertificate -FilePath $RutaExportacion '
    -CertStoreLocation "cert:\LocalMachine\Root" -Password $Password

# Importar certificado PFX TrustedPublisher
Import-PfxCertificate -FilePath $RutaExportacion '
    -CertStoreLocation "cert:\LocalMachine\TrustedPublisher" -Password
    $Password

```

Listado código 7.16: Código CreacionCertificadoAutofirmado.ps1

Una vez se ha generado el certificado autofirmado, se va a proceder a firmar un *script* llamada "Fecha1.ps1". Para ello se ha utilizado el código mostrado en el listado de código 7.17.

```

<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Este script firma un script con
    un certificado autofirmado
#>

$MiCertificado = Get-PfxCertificate -FilePath "E:\Certificados\
    MiCertificadoFirma.pfx"

```


7.1.4.3. EventLog

Todo administrador de sistemas sabe que cada vez que sucede algo en Windows se guarda en el Registro de Eventos. Por defecto existe cuatro tipos de registro de eventos predeterminados que son:

- **Aplicación.** Muestra eventos relacionados con el software instalado en un equipo.
- **Seguridad.** Contiene eventos relacionados con la seguridad del equipo.
- **Instalación.** Contiene eventos de configuración relacionados con instalaciones de KB (actualizaciones de Microsoft), eventos de rendimiento y eventos que ocurren durante la instalación.
- **Sistema.** Muestra eventos relacionados con el sistema: reinicio del sistema, reinicio de servicios, archivos del sistema, etc.).

Cada evento consta de múltiples campos estándar. Estos campos son los siguientes:

- **Nombre de registro.** Es el nombre del registro de eventos que se desea ver. Son: aplicación, seguridad, sistema, etc..
- **Fuente.** En un nombre que le permite distinguir la fuente de los eventos. Por lo general, será un nombre de aplicación o servicio que creó un evento.
- **ID de evento.** Es un identificador de evento. Cada ID de evento cuenta una historia diferente. Combinado con el nombre de registro es una de las informaciones más importantes.
- **Nivel.** Hay cinco niveles disponibles: detallado, informativo, advertencia, error y crítico. A menudo, al diagnosticar problemas, se puede filtrar los registros solo a nivel crítico y de error para obtener información instantánea sobre un problema en la estación de trabajo o en el servidor.
- **Usuario.** Generalmente contiene un usuario que generó un evento. Por lo general será SYSTEM o N/A (No disponible).
- **Registrado.** Hora en que se inició la sesión del evento.
- **Categoría de tarea.** Cada origen de evento puede tener su categoría. Algunos eventos lo tienen, otros no.
- **Palabras clave.** Hay varias pero las más importantes son: auditoría correcta y error de auditoría.
- **Equipo.** Nombre del equipo que informó el evento.

Pero no son los únicos. Cada evento también consta de un mensaje y muchos otros campos que contienen datos. Depende de los desarrolladores elegir cómo quieren construir sus eventos y qué propiedades llenan. Esto dificulta un poco la automatización porque cada tipo de evento puede tener una estructura completamente diferente (hasta cierto punto).

Con PowerShell se disponen de muchos cmdlets para acceder a los eventos generados en el equipo. Se ha implementado un pequeño *script* para obtener la información de los usuarios que han iniciado y cerrado sesión en uno o varios equipos desde un número determinado de días. El código se puede ver en el listado de código 7.18.

```

<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Se muestra los inicios y cierres de sesi n en un equipo.

.EXAMPLE
    Get-InfoSesionesEquipos -nombreEquipo 'Equipo' -NumeroDias 15

.PARAMETER NombreEquipo
    Equipo sobre el que se quiere obtener informacion.

.PARAMETER NumeroDias
    Numero de dias sobre los que se quiere obtener informacion.
#>

function Get-InfoSesionesEquipos {
    [CmdletBinding()]
    param(
        [Parameter(Mandatory = $true ,
            Position = 1,
            ValueFromPipeline = $true ,
            ValueFromPipelineByPropertyName = $true)]
        [ValidateNotNullOrEmpty()]
        [string []]$nombreEquipo,

        [Parameter(Mandatory = $true)]
        [string]$NumeroDias
    )

    BEGIN {
        ForEach ($equipo in $nombreEquipo) {
            # Compruebo que hay conexion con el equipo
            if (Test-Connection $equipo -Quiet -Count 1) {
                if ((Get-Service -ComputerName $equipo -Name RemoteRegistry
                    ).Status -ne 'Running') {
                    # Es necesario que el servicio RemoteRegistry est
                    # arrancado. Se hace.
                    Set-Service -ComputerName $equipo -Name RemoteRegistry
                        -StartupType Automatic -Status Running -Verbose
                }
            }
        }
    }

    PROCESS {
        $res = @()
        ForEach ($equipo in $nombreEquipo) {
            # Compruebo que hay conexion con el equipo
            if (Test-Connection $equipo -Quiet -Count 1)
            {
                # inicio y cierre sesion local
                $logs = Get-EventLog System -ComputerName $equipo -Source
                    Microsoft-Windows-Winlogon -After (Get-Date).AddDays(-(
                    $NumeroDias -as [int]))
                if ($logs.Count -ne 0) {
                    ForEach ($log in $logs) {
                        if($log.instanceid -eq 7001) {
                            $type = "Inicia Sesion"
                        } Elseif ($log.instanceid -eq 7002) {

```



```

PS D:\FeoIde\6_TFG\1_Scripts\Creaciones> Get-WinEvent -FilterHashtable @{
    LogName = 'Application'
    ProviderName = '*defrag'
}

ProviderName: Microsoft-Windows-Defrag

TimeCreated      Id LevelDisplayName Message
-----
25/08/2019 12:57:47 258 Información El optimizador de almacenamiento completó correctamente desfragmentación en DISCO_3TB (D:)
25/08/2019 12:56:49 258 Información El optimizador de almacenamiento completó correctamente desfragmentación en SISTEMA W10 SSD (C:)
25/08/2019 12:56:49 258 Información El optimizador de almacenamiento completó correctamente volver a optimizar en SISTEMA W10 SSD (C:)
25/08/2019 12:48:00 258 Información El optimizador de almacenamiento completó correctamente desfragmentación en DISCO_500GB (E:)
25/07/2019 12:55:27 258 Información El optimizador de almacenamiento completó correctamente desfragmentación en DISCO_3TB (D:)
25/07/2019 12:55:15 258 Información El optimizador de almacenamiento completó correctamente desfragmentación en SISTEMA W10 SSD (C:)
25/07/2019 12:55:15 258 Información El optimizador de almacenamiento completó correctamente volver a optimizar en SISTEMA W10 SSD (C:)
25/07/2019 12:47:54 258 Información El optimizador de almacenamiento completó correctamente desfragmentación en DISCO_500GB (E:)
26/06/2019 10:05:57 258 Información El optimizador de almacenamiento completó correctamente desfragmentación en DISCO_3TB (D:)
26/06/2019 10:04:34 258 Información El optimizador de almacenamiento completó correctamente desfragmentación en SISTEMA W10 SSD (C:)
26/06/2019 10:04:34 258 Información El optimizador de almacenamiento completó correctamente volver a optimizar en SISTEMA W10 SSD (C:)
26/06/2019 10:01:57 258 Información El optimizador de almacenamiento completó correctamente desfragmentación en DISCO_500GB (E:)

```

Figura 7.34: Creación de consultas Get-WinEvent con FilterHashtable

Por último y para ir acabando este apartado, el propio PowerShell tiene una herramienta muy potente que es el envío de la salida de cualquier cmdlet de PowerShell a una tabla interactiva que se muestra en una ventana independiente. Lanzando `Get-EventLog -LogName System | Out-GridView` se muestra la siguiente ventana a la que podemos realizar multitud de filtros de un modo gráfico tal como se observa en la figura 7.35.

Index	Time	EntryType	Source	InstanceID	Message
353.421.125	31/08/2019 23:36:40	Warning	DCOM	10.016	No se encontró la descripción del id. de evento
353.421.124	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.472	El servicio Kaspersky Security Network proxy s %%1069
353.421.123	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.503	El servicio Kaspersky Security Network proxy s El servicio ksnproxy no se pudo iniciarse como %%1326
353.421.122	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.510	Para asegurarse de que el servicio esté correct
353.421.121	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.472	El servicio Kaspersky Security Network proxy s %%1069
353.421.120	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.503	El servicio Kaspersky Security Network proxy s El servicio ksnproxy no se pudo iniciarse como %%1326
353.421.119	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.510	Para asegurarse de que el servicio esté correct
353.421.118	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.472	El servicio Kaspersky Security Network proxy s %%1069
353.421.117	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.503	El servicio Kaspersky Security Network proxy s El servicio ksnproxy no se pudo iniciarse como %%1326
353.421.116	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.510	Para asegurarse de que el servicio esté correct
353.421.115	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.472	El servicio Kaspersky Security Network proxy s %%1069
353.421.114	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.503	El servicio Kaspersky Security Network proxy s El servicio ksnproxy no se pudo iniciarse como %%1326
353.421.113	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.510	Para asegurarse de que el servicio esté correct
353.421.112	31/08/2019 23:00:41	Error	Service Control Manager	3.221.232.472	El servicio Kaspersky Security Network proxy s %%1069

Figura 7.35: Visualización de Out-GridView para una consulta de eventos en PowerShell

Mediante PowerShell, el administrador de sistemas puede hacer consultas a los logs de eventos del sistema de una manera muy sencilla y realizar pequeños códigos en lenguaje de *scripting* para obtener cuanta información necesite para descubrir problemas y

anomalías de un equipo y poder planificar una parada programada para solucionar la incidencia y conseguir restablecer el funcionamiento adecuado en ese equipo.

7.1.4.4. Máquinas virtuales Hyper-V (VM)

En la actualidad en los departamentos se disponen de muchas máquinas virtuales para su funcionamiento. En concreto se suelen tener virtualizados los equipos que tienen las aplicaciones de licencias de algún software comprado por el departamento. Las ventajas de emplear máquinas virtuales son muchas pero quedan fuera de este trabajo por lo que no se van a mencionar. Sin embargo, desde PowerShell se puede gestionar fácilmente las máquinas virtuales de Hyper-V. En nuestro departamento disponemos de varias máquinas con Hyper-V por lo que disponer un *script* para su gestión es muy útil y sencillo.

```
<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Comandos en PowerShell muy interesantes para gesti n de Hyper-V.

#>
# Se establece conexi n con la m quina que contiene las m squinas
  virtuales de Hyper-V
Enter-PSSession -ComputerName $nombreEquipo

# Mostrar los comandos del m dulo Hyper-V
Get-Command -Module Hyper-V

# Actualizar la ayuda del m dulo Hyper-V
Update-Help -Module Hyper-V

# Informaci n de la m quina contenedora de Hyper-V
# (nombre equipo, Numero procesadores logicos, memoria, si la migracion
  esta activa)
Get-VMHost

# Muestra las maquinas virtuales
Get-VM

# Arrancar una maquina virtual
Start-vm -name "nombre maquina virtual"

# Parar una maquina virtual
Stop-vm -name "nombre maquina virtual"

# Apagar todas las maquinas virtuales con confirmacion
Get-VM | where {$_.State -eq 'Running'} | Stop-VM -Confirm

# Muestra informaci n de las maquina virtuales que est n en ejecucion
Get-VM | where {$_.state -eq 'running'} | sort Uptime | select Name,@{N="
  Tiempo Encendido";E={$_.Uptime}}, '
  ProcessorCount,@{N="MemoryMB";E={$_.MemoryAssigned/1MB}}, Status,
  Generation,Path, '
  IntegrationServicesState | Format-Table AutoSize

# Crear un punto de control de una maquina virtual
```

```

Get-VM -Name "nombre maquina virtual" | Checkpoint-VM -SnapshotName "nombre
para snapshot" -verbose

# Ver las instanteneas asociadas a una maquina virtual
Get-VMSnapshot -VMName "nombre maquina virtual"

# Eliminar una maquina virtual
Remove-VM -Name "nombre maquina virtual" -Force -Confirm

# Mover una maquina virtual
Move-VMStorage "nombre maquina virtual" -DestinationStoragePath "Ruta
destino"

```

Listado código 7.19: Código ComandosHyperV.ps1

En el listado de código 7.19, se dispone de unos cuantos cmdlets de PowerShell para gestionar nuestras máquinas virtuales. Lo primero que se ha de hacer es establecer una conexión con la máquina que alberga Hyper-V. Se pueden mostrar las máquinas virtuales, pararlas y encenderlas, actualizar la ayuda de Hyper-V que viene muy bien para afinar el código que se va generando para la administración de los sistemas. Se puede eliminar una máquina virtual, mover de ubicación la máquina virtual, etc.. Crear una instantánea (snapshot) de la máquina se realiza con un comando muy sencillo (si se tuviera que realizar gráficamente estas operaciones se tendrían que hacer numerosos «clicks» en la interfaz gráfica). También se puede de un modo muy sencillo ver las instantáneas asociadas a una máquina virtual. En [54] se dispone de información de cómo trabajar con Hyper-V mediante PowerShell.

Se ha implementado un sencillo *script* para realizar una copia de seguridad de nuestras máquinas virtuales porque no están libres de fallo. Es por ello, que es fundamental tener copias de nuestras máquinas virtuales ya que si se corrompieran y no tuviésemos copia, sería un desastre y se necesitaría bastante tiempo para restablecer el servicio. ¡No es deseable nunca! y por ello se realizan copias periódicamente.

```

<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESCRIPTION
    Se realiza una exportacion de las maquina virtuales.
    Se emplea para copias de seguridad de las VM.
    Se envia un correo electronica al acabar
    Requiere lanzarse en la consola como administrador

#>

# Carpeta donde se va a realizar el backup
$Backup = "e:\BACKUP\AIMSRVAULAS"

# Se obtienen las maquinas virtuales
$VMs = Get-VM | select -Property Name

# Se comprueba que la carpeta de backup existe. Si no... se crea
if (!(Test-Path -Path $Backup)) {
    New-Item $Backup -type directory -force

```



```

}

$correcto = $false
# Se hace una exportacion de las maquinas virtuales
foreach ($VM in $Vms) {
    try {
        Write-Verbose "Se va a proceder a exportar la maquina virtual $VM"
        Export-VM -Name $VM.name -Path $Backup -ErrorAction stop
        $correcto = $true
    } catch {
        Write-Warning "No se ha podido exportar la maquina virtual $VM"
        $correcto = $false
    }
}

# Se envia un correo electronico
if ($correcto) {
    try {
        Send-MailMessage -From "Backup_VM_AIM@upv.es" -To "fedelolmeda@upv.es" -Subject "Se ha realizado una copia de las VM de Hyper-V " -Body "$($Get-Date), se ha realizado una copia de seguridad en la ruta $Backup" -SmtpServer "smtp.upv.es" -errorAction Stop
    } catch {
        Write-Warning "No ha sido posible enviar el correo de exportacion de las VM"
    }
} else {
    Write-Warning "Ha habido un error y no se ha hecho nada"
}

```

Listado código 7.20: Código ExportarVM.ps1

En el listado de código 7.20 se muestra el código para la realización de la exportación de las máquinas virtuales. Se ha probado la exportación sobre un NAS y sobre una carpeta compartida pero se produce un error. Probablemente tenga que ver con el protocolo SAMBA y se ha revisado en algunos foros en Internet y al parecer es común este problema y no hay documentada ninguna solución. Por el momento se realiza la exportación a un disco duro local de la máquina que contiene las máquinas virtuales aunque se continuará investigando este punto con el fin de poder exportar a un recurso compartido (ya sea en un NAS o en una máquina física).

Finalmente solo queda implementar un *script* para crear una tarea programada, que se ejecute periódicamente y de modo transparente, la exportación de las máquinas virtuales que se disponen en el AIM. El código se puede ver en el listado de código 7.21.

```

<#
.NOTES
    Creado por:      Fede Olmeda
    Creado en:      @2019

.DESRIPTION
    Se crea una tarea programada para lanzar un script en PowerShell.

#>

```

```
# Ruta del script de PowerShell
$script1 = "E:\Scripts\ExportarVM.ps1"
# Se va a ejecutar una vez al mes
$trigger = New-JobTrigger -Weekly -DaysOfWeek Sunday -WeeksInterval 4 -At
00:30AM
$options = New-ScheduledJobOption -RunElevated

#Se crea la tarea programada
Register-ScheduledJob -Name "Exportar VM mensual" -FilePath $script1 -
Trigger $trigger -ScheduledJobOption $options
```

Listado código 7.21: Código TareaProgramadaExportarVM.ps1

7.1.5. Resumen *scripts* desarrollados

En la tabla 7.5 se muestra los scripts que he desarrollado para este trabajo junto con una breve explicación de la funcionalidad que realizan. La mayoría de ellos se han implementado con la finalidad de poder automatizar tareas rutinarias de administración de sistemas que se han de realizar con mucha frecuencia para un correcto mantenimiento del parque de equipos informáticas que dispone el AIM.

Se puede observar que se han desarrollado unas cuantas funciones en ficheros de PowerShell independientes. La justificación de haber creado cada función independientemente se debe a dos motivos:

1. Es mucho más sencillo poder utilizar estas funciones en la creación de nuevas tareas.
2. Sencillez a la hora de depurar las funciones.

Es muy interesante esta forma de trabajo modular para la creación de otras funciones o tareas mucho más complejas y sofisticadas.

Nombre del Script	Descripción de su funcionalidad
Habilitar-PsRemoto.ps1	Permite el acceso remoto (con PS) a un equipo al que no se puede acceder.
Get-InfoInformacionEquipo.ps1	Obtiene información personalizada (principalmente de hardware) de uno o varios equipos guardando en un fichero el del equipo que ha generado un error.
Get-InfoInfoSO.ps1	Muestra información personalizada del sistema operativo de un equipo (versión SO, servicepack, etc.).
Get-InfoInfoSistema.ps1	Muestra información personalizada de un equipo (fabricante, modelo, memoria, etc.).
Get-InfoInfoDisco.ps1	Muestra información personalizada de los discos duros de un equipo.
Get-InfoInfoProceso.ps1	Muestra información personalizada de los procesos de un equipo (nombre del proceso y su ejecutable).
Get-InfoInfoServicioRevision.ps1	Muestra información personalizada de los servicios de un equipo que cumplen que estén en ejecución y tengan el modo de inicio en automático.
Get-InfoInfoRed.ps1	Muestra información personalizada de la red de un equipo.
New-InformeInformeHTML.ps1	Genera en formato HTML un archivo por cada equipo sobre el que se va actuar en una ruta que se le ha especificado como parámetro. Realiza llamadas a otras funciones que se han desarrollado para la generación del archivo HTML.
Get-MonitorMonitorServidores.ps1	Genera un archivo en formato HTML con información personalizada de uno o varios equipos. Dicho archivo se guarda en una ruta en el disco duro y se envía un correo electrónico al administrador de los equipos adjuntándole el archivo generado.
TareaMonitorMonitorServidores.ps1	Genera una tarea programada del script Get-MonitorMonitorServidores para que diariamente se ejecute la tarea programada a una hora determinada.
Get-BackUpBackUp.ps1	Realiza una copia de seguridad con Robocopy enviando un correo electrónico al administrador de sistemas al finalizar. Se adjunta el archivo log que se ha generado y que se ha almacenado en una ubicación determinada.
Clean-ArchivosArchivosTemporales.ps1	Realiza un borrado de archivos temporales en un equipo o varios.
Get-FechaFechaEncendidoEquipo.ps1	Indica cuándo se ha encendido un equipo pasado como parámetro.
CreacionCertificadoAutofirmadoAutofirmado.ps1	Genera un certificado autofirmado para poder firmar scripts realizados en PS.
FirmaScriptFechaFecha.ps1	Firma un script creado con un certificado autofirmado creado previamente.
Get-InfoInfoSesionesEquiposEquipos.ps1	Muestra los inicios y cierre de sesión de un equipo o varios en los últimos X días (donde Xse es un parámetro que se le ha de indicar a la función).
ComandosComandosHyperVHyperV.ps1	Conjunto de comandos en PS para gestionar Hyper-V (mostrar máquinas, arrancar y parar VM, crear instantánea, etc.).
ExportarExportarVMVM.ps1	Realiza una exportación de unas máquinas virtuales a una carpeta local enviando un correo electrónico al administrador de sistemas al hacerlo.
TareaProgramadaProgramadaExportarVMExportarVM.ps1	Realiza una tarea programada del script exportarVM.ps1

Tabla 7.5: Tabla resumen de los *scripts* que se han desarrollado en este trabajo

CAPÍTULO 8

Conclusiones

Se ha conseguido los objetivos propuestos en este TFG y que paso a enumerar:

- He empleado herramientas gratuitas para dar cabida a las necesidades actuales que se tiene en el Área de Ingeniería Mecánica (AIM).
- He implementado mediante OpenGnsys una solución adecuada para su utilización en los equipos informáticos que se disponen en el Área de Ingeniería Mecánica (AIM).
 - He creado desde cero un servidor de OpenGnsys para la gestión de los equipos a administrar.
 - He creado un repositorio que va a albergar las imágenes necesarias para la adecuada gestión de los equipos del AIM.
 - He definido una unidad organizativa (AIM) estructurada para definir ámbitos de aplicación.
 - He definido nuevos procedimientos en OpenGnsys para gestionar adecuadamente los equipos.
 - He modificado archivos de configuración para adaptarlo a las necesidades requeridas.
 - He instalado un certificado para el acceso seguro a la consola Web de OpenGnsys.
 - He realizado las modificaciones necesarias para el funcionamiento de RemotePC que se integra en OpenGnsys junto con el servicio que ofrece la UPV para que alumnos que realizan prácticas en nuestras aulas puedan acceder a equipos remotamente fuera de las horas lectivas.
 - He creado una imagen base con lo necesario para su utilización en las aulas docentes del AIM.
 - He realizado el despliegue de una imagen base previamente creada a los diferentes equipos que conforman las aulas docentes del AIM.
 - He aplicado adecuadamente la configuración deseada sobre los equipos de las aulas docentes que se han clonado (cambio de nombre, meter en el dominio, antivirus, etc.). Se dispone en todos los equipos de las diferentes aulas que hay en el AIM de la misma configuración.
- He mejorado tanto mis conocimientos de PowerShell como mis habilidades de *scripting*.
- He administrado equipos desde la línea de comandos PowerShell.

- He implementado *scripts* utilizando PowerShell que permiten recuperar remotamente información de todo tipo de los equipos (de aulas, servidores, etc.), saber los usuarios que inician y cierran sesión, saber cuándo se ha encendido un equipo, acceso a los eventos de los equipos, etc. que da mucha información al administrador del sistema para la gestión diaria de los equipos que ha de mantener.
- He creado un certificado autofirmado para poder firmar los *scripts* que se generen y ser ejecutados con seguridad.
- He implementado utilizando PowerShell un *script* para la gestión de máquinas virtuales bajo Hyper-V que dispone el AIM (principalmente servidores de licenciamiento de aplicaciones compradas por el AIM).
- He implementado *scripts* desarrollados en PowerShell para automatizar tareas de administración de sistemas.
 - He implementado informes en HTML muy completos con información muy útil de los equipos del AIM.
 - He implementado un *script* utilizando PowerShell para la realización de una copia de seguridad de los datos de un equipo y enviando posteriormente un correo electrónico al administrador de sistema de la realización de la tarea de copia de seguridad.
 - He implementado un *script* utilizando PowerShell para la limpieza de archivos temporales en los equipos del AIM.
 - He implementado varios *scripts* utilizando PowerShell para crear una tarea programada de una función (también realizar en PowerShell).

En definitiva, he implementado *scripts* utilizando PowerShell para automatizar tareas de administración de sistemas y para una mejor gestión de los equipos que he de administrar en el AIM.

Llevo muchos años trabajando en la UPV como técnico de laboratorio en el AIM y este TFG me ha permitido abrir la mente hacia una mejor administración de los equipos a mi cargo. He trabajado con OpenGnsys y comprobado su potencial a la hora de realizar la clonación de equipos. Es una herramienta en constante evolución con multitud de opciones para que se pueda adaptar a una amplia variedad de requisitos. El hecho de disponer de una API de funciones te permite realizar nuevas funciones no implementadas que se puede ajustar muchísimo a tus necesidades. Se necesita mucho más tiempo para poder explotar al límite las posibilidades de esta herramienta que me ha sorprendido gratamente. Respecto a PowerShell, decir que es una herramienta excepcional que da mucho juego para la administración de sistemas. Tengo claro que hay que automatizar muchos procesos rutinarios en el día a día y PowerShell es una herramienta fantástica para ello. Con él puedes acceder a todo en los sistemas Windows pero para ello necesitas mucho tiempo (para mejorar tus conocimientos y habilidades y realizar pruebas antes de ponerlo en producción). Como he comentado en la introducción de este TFG, el trabajo realizado no es una solución completa sino un inicio hacia una administración más eficiente, segura y tolerante a fallos de los equipos que se han de administrar.

Habiéndose alcanzado los objetivos propuestos para la realización de este TFG se plantean algunas propuestas de mejora:

-
- Ampliación de los conocimientos de la API de funciones del motor de clonación de OpenGnsys. Se implementarían nuevas funciones que mejorarían la gestión y el desarrollo para realizar un despliegue de una imagen base sobre un ámbito de aplicación mediante OpenGnsys.
 - Implementación de nuevos procedimientos en OpenGnsys para la gestión más eficiente de los equipos que se administran.
 - Implementación de nuevas tareas en OpenGnsys orientadas a mejorar la automatización de los procesos que se necesitan en la gestión de los equipos a administrar. A pesar que se suelen realizar una o dos veces al año la clonación de los equipos con imágenes base actualizadas, siempre es conveniente mejorar los procesos de automatización para minimizar tiempos y errores de ejecución.
 - Creación de una interfaz gráfica (GUI) y dotarla de funcionalidad a través de la realización de un conjunto de *scripts* realizados mediante PowerShell.
 - Ampliar notablemente las tareas y procedimientos a realizar con PowerShell. Los requerimientos de administración de los equipos no son estáticos y hay que adaptarse a las nuevas necesidades.

Para concluir indicar que la realización de esta memoria se ha realizado en L^AT_EX. Me ha sorprendido su potencia y el resultado es espectacular.

Bibliografía

- [1] Symantec Ghost Solution Suite. [en línea]. 2019. [Consulta: 1 julio 2019]. Disponible en: Hoja de datos de Symantec Ghost Solution Suite
- [2] Acronis Snap Deploy. [en línea]. 2019. [Consulta: 1 julio 2019]. Disponible en: Hoja de datos de Acronis Snap Deploy
- [3] SmartDeploy. [en línea]. 2019. [Consulta: 1 julio 2019]. Disponible en: SmartDeploy
- [4] CloneDeploy. [en línea]. 2019. [Consulta: 1 julio 2019]. Disponible en: CloneDeploy
- [5] Kace SDA. [en línea]. 2019. [Consulta: 1 julio 2019]. Disponible en: KACE Systems Deployment Appliance
- [6] Fog Project. [en línea]. 2019. [Consulta: 1 julio 2019]. Disponible en: Fog Project
- [7] Introducción a Microsoft Deployment Toolkit (MDT). [en línea]. 2019. [Consulta: 5 julio 2019]. Disponible en: Introducción a Microsoft Deployment Toolkit (MDT)
- [8] RAE. Clonar. [en línea]. 2019. [consulta: 1 agosto 2019]. Disponible en: <https://dle.rae.es/srv/search?m=30&w=clonar>
- [9] DOBLAS VISO, ANTONIO J. (2018). Curso básico de OpenGnsys 1.1.0: Tema 6.2 Restauración y Despliegue de imágenes. 20/01/2018, de OpenGnsys. [en línea]. 2019. [consulta: 10 junio 2019]. Disponible en: www.opengnsys.es
- [10] OpenGnsys. [en línea]. 2019. [consulta: 27 mayo 2019]. Disponible en: <https://opengnsys.es/web/>
- [11] WIKIPEDIA. Unidifusión. [en línea]. 2019. [consulta: 27 mayo 2019]. Disponible en: <https://es.wikipedia.org/wiki/Unidifusi3n>
- [12] WIKIPEDIA. Multidifusi3n. [en línea]. 2019. [consulta: 27 mayo 2019]. Disponible en: <https://es.wikipedia.org/wiki/Multidifusi3n>
- [13] WIKIPEDIA. Peer-to-peer. [en línea]. 2019. [consulta: 27 mayo 2019]. Disponible en: <https://es.wikipedia.org/wiki/Peer-to-peer>
- [14] WIKIPEDIA. Hardware. [en línea]. 2019. [consulta: 27 mayo 2019]. Disponible en: <https://es.wikipedia.org/wiki/Hardware>
- [15] WIKIPEDIA. Software. [en línea]. 2019. [consulta: 27 mayo 2019]. Disponible en: <https://es.wikipedia.org/wiki/Software>
- [16] Instalaci3n de OpenGnsys Server. [en línea]. 2019. [consulta: 27 mayo 2019]. Disponible en: <https://opengnsys.es/trac/wiki/ManualInstalacion>

- [17] Lista de compatibilidad para la instalación de OpenGnsys Server. [en línea]. 2019. [consulta: 27 mayo 2019]. Disponible en: <https://www.opengnsys.es/trac/wiki/CompatibilidadInstalacionServer>
- [18] WIKIPEDIA. Imagen ISO. [en línea]. 2019. [consulta: 27 mayo 2019]. Disponible en: https://es.wikipedia.org/wiki/Imagen_ISO
- [19] Repositorio CICA. [en línea]. 2019. [consulta: 15 mayo 2019]. Disponible en: <https://ubuntu.cica.es/releases/xenial/>
- [20] Rufus. [en línea]. 2019. [consulta: 15 mayo 2019]. Disponible en: <https://rufus.ie/>
- [21] Configuración del ordenador para arrancar desde USB. [en línea]. 2019. [consulta: 15 mayo 2019]. Disponible en: <https://www.osi.es/es/desinfecta-tu-ordenador/configuracion-ordenador-arrancar-cd-dvd-usb-rescate>
- [22] WIKIPEDIA. Secure Shell. [en línea]. 2019. [consulta: 15 mayo 2019]. Disponible en: https://es.wikipedia.org/wiki/Secure_Shell
- [23] MobaXterm. [en línea]. 2019. [consulta: 15 mayo 2019]. Disponible en: <http://mobaxterm.mobatek.net/download.html>
- [24] WIKIPEDIA. Nombre de equipo. [en línea]. 2019. [consulta: 15 mayo 2019]. Disponible en: https://es.wikipedia.org/wiki/Nombre_de_equipo
- [25] Curso básico OpenGnsys 1.1.0. [en línea]. 2019. [consulta: 15 abril 2019]. Disponible en: <https://docencia-net.cv.uma.es/>
- [26] DOBLAS VISO, ANTONIO JESÚS. GÓMEZ LABRADOR, RAMÓN. (2018). Curso básico de OpenGnsys 1.1.0: Tema 2 Instalación de OpenGnsys. 09/05/2018, de OpenGnsys. [en línea]. 2019. [consulta: 16 abril 2019]. Disponible en: www.opengnsys.es
- [27] WIKIPEDIA. Subversion (software). [en línea]. 2019. [consulta: 16 abril 2019]. Disponible en: [https://es.wikipedia.org/wiki/Subversion_\(software\)](https://es.wikipedia.org/wiki/Subversion_(software))
- [28] WIKIPEDIA. Samba (software). [en línea]. 2019. [consulta: 1 junio 2019]. Disponible en: [https://es.wikipedia.org/wiki/Samba_\(software\)](https://es.wikipedia.org/wiki/Samba_(software))
- [29] Autoridad de certificación gratuita. Let's Encrypt. [en línea]. 2019. [consulta: 1 junio 2019]. Disponible en: <https://letsencrypt.org/es/>
- [30] Solicitud entregada certificado servidor. [en línea]. 2019. [Consulta: 1 junio 2019]. Disponible en: <http://www.upv.es/doc/certificado/entregada>
- [31] WIKIPEDIA. Servidor HTTP Apache. [en línea]. 2019. [consulta: 1 junio 2019]. Disponible en: https://es.wikipedia.org/wiki/Servidor_HTTP_Apache
- [32] DOBLAS VISO, ANTONIO JESÚS. (2018). Curso básico de OpenGnsys 1.1.0: Tema 3.1 Administración de componentes. 18/04/2018, de OpenGnsys. [en línea]. 2019. [consulta: 16 abril 2019]. Disponible en: www.opengnsys.es
- [33] DOBLAS VISO, ANTONIO JESUS. (2018). Curso básico de OpenGnsys 1.1.0: Tema 3.3 Administración de una unidad organizativa. 26/04/2018, de OpenGnsys. [en línea]. 2019. [Consulta: 17 abril 2019]. Disponible en: www.opengnsys.es
- [34] WIKIPEDIA. Servidor proxy. [en línea]. 2019. [consulta: 15 junio 2019]. Disponible en: https://es.wikipedia.org/wiki/Servidor_proxy

-
- [35] Proyectos con OpenGnsys: Remote PC. [en línea]. 2019. [consulta: 12 junio 2019]. Disponible en: <https://www.udsenderprise.com/es/blog/2017/09/14/proyectos-con-opengnsys-remote-pc/>
- [36] VirtualCable apuesta por OpenGnsys para el acceso a PCs remotos. [en línea]. 2019. [consulta: 12 junio 2019]. Disponible en: <https://www.udsenderprise.com/es/blog/2019/02/14/virtualcable-apuesta-por-opengnsys-para-el-acceso-/>
- [37] Proyecto RemotePC: acceso remoto a laboratorios TIC. [en línea]. 2019. [consulta: 20 junio 2019]. Disponible en: <https://opengnsys.es/trac/wiki/RemotePc>
- [38] ALONSO MARTINEZ, JONATHAN. DOBLAS VISO, ANTONIO J. (2018). Curso básico de OpenGnsys 1.1.0: Tema 5 Particionado de discos. 09/05/2018, de OpenGnsys. [en línea]. 2019. [Consulta: 18 abril de 2019]. Disponible en: www.opengnsys.es
- [39] WIKIPEDIA. Partición de disco. [en línea]. 2019. [consulta: 10 julio 2019]. Disponible en: https://es.wikipedia.org/wiki/Partición_de_disco
- [40] WIKIPEDIA. Tabla de particiones GUID. [en línea]. 2019. [consulta: 10 julio 2019]. Disponible en: https://es.wikipedia.org/wiki/Tabla_de_particiones_GUID
- [41] DOBLAS VISO, ANTONIO JESÚS. GÓMEZ LABRADOR, RAMÓN. (2018). Curso básico de OpenGnsys 1.1.0: Tema 6.1 Creación de imágenes. 10/05/2018, de OpenGnsys. [en línea]. 2019. [Consulta: 17 abril 2019]. Disponible en: www.opengnsys.es
- [42] DOBLAS VISO, ANTONIO J. (2018). Curso básico de OpenGnsys 1.1.0: Tema 6.2 Restauración y Despliegue de imágenes. 10/05/2018, de OpenGnsys. [en línea]. 2019. [Consulta: 17 abril 2019]. Disponible en: www.opengnsys.es
- [43] Wikipedia. Script. [en línea]. 2019, [consulta: 1 septiembre 2019]. Disponible en: <https://es.wikipedia.org/wiki/Script>
- [44] Musy, Julien, and ENI Biblioteca Online. Windows PowerShell [Recurso electrónico-En línea]: administrar puestos cliente Windows. ENI, 2018. [Herramienta Polibuscador Biblioteca UPV]. 2019. [Consulta: 10 agosto 2019]. Disponible en: Windows PowerShell: administrar puestos cliente Windows
- [45] Wilson, Ed. Windows PowerShell 3.0 Step by Step. Microsoft Press, 2013. [Herramienta Polibuscador Biblioteca UPV]. 2019. [Consulta: 12 mayo 2019]. Disponible en: Windows PowerShell 3.0 Step by Step
- [46] Jones, Don, and Jeffery Hicks. Learn Windows PowerShell in a Month of Lunches. Third edition., 2017. [Herramienta Polibuscador Biblioteca UPV]. 2019. [Consulta: 8 julio 2019]. Disponible en: Learn Windows PowerShell in a month of lunches
- [47] Wikipedia. Powershell. [en línea]. 2019. [Consulta: 1 agosto 2019]. Disponible en: <https://en.wikipedia.org/wiki/PowerShell>
- [48] Instalación y uso de Windows PowerShell Web Access. [en línea]. 2019. [Consulta: 1 agosto 2019]. Disponible en: <https://docs.microsoft.com/es-es/powershell/scripting/components/web-access/install-and-use-windows-powershell-web-access?view=powershell-6#install-windows-powershell-web-access-using-powershell-cmdlets>

- [49] Microsoft Virtual Machine Bus (VMBus). [en línea]. 2019. [Consulta 1 agosto 2019]. Disponible en <https://searchwindowsserver.techtarget.com/definition/Microsoft-Virtual-Machine-Bus-VMBus>
- [50] Iniciar sesión como proceso por lotes. [en línea]. 2019. [Consulta: 20 agosto 2019]. Disponible en: <https://docs.microsoft.com/es-es/windows/security/threat-protection/security-policy-settings/log-on-as-a-batch-job>
- [51] Robocopy. [en línea]. 2019. [Consulta: 1 agosto 2019]. Disponible en: <https://docs.microsoft.com/es-es/windows-server/administration/windows-commands/robocopy>
- [52] Robocopy Exit Codes. [en línea]. 2019. [Consulta: 1 agosto 2019]. Disponible en: <https://ss64.com/nt/robocopy-exit.html>
- [53] Ccleaner. [en línea]. 2019. [Consulta: 1 junio 2019]. Disponible en: Herramienta limpieza PC Ccleaner
- [54] Hyper-V. [en línea]. 2019. [Consulta: 5 agosto 2019]. Disponible en: Hyper-V con PowerShell
- [55] Creación de consultas Get-WinEvent con FilterHashtable. [en línea]. 2019. [Consulta: 15 agosto 2019]. Disponible en: Consultas Get-WinEvent

Glosario

- API** Application Programming Interface. 46
- ASIC** Área de Sistemas de la Información y las Comunicaciones. 60
- BIOS** Basic Input/Output System. 25
- CPU** Central Processing Unit. 29
- CSR** Certificate Signing Request. 38
- DHCP** Dynamic Host Configuration Protocol. 26
- DNS** Domain Name System. 57
- FQDN** Fully Qualified Domain Name. 29
- IP** Internet Protocol. 29
- LVM** Logical Volume Manager. 31
- MAC** Media Access Control. 51
- MBR** Master Boot Record. 65
- NAS** Network Attached Storage. 149
- NTP** Network Time Protocol. 57
- OU** Organizational Unit. 52
- P2P** Peer-to-Peer. 25
- PXE** Preboot Execution Environment. 33
- RAE** Real Academia Española. 19
- RAID** Redundant Array of Independent Disks. 73
- RDP** Remote Desktop Protocol. 61
- SATA** Serial Advanced Technology Attachment. 26
- SID** Security IDentifier. 21

SSH Secure Shell. 28

TIC Tecnologías de la Información y la Comunicación. 23

UPV Universitat Politècnica de València. 40

URL Uniform Resource Locator. 35

USB Universal Serial Bus. 25

WOL Wake On Lan. 63