Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Análisis e implantación de mejoras para el Laboratorio de Telemática

Autor: Daniel Masero Berzal

Tutor: Francisco Javier Muñoz Calle

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020

Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Análisis e implantación de mejoras para el Laboratorio de Telemática

Autor:
Daniel Masero Berzal

Tutor:
Francisco Javier Muñoz Calle
Profesor adscrito

Dpto. de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2020

Proyecto Fin de Carrera: Análisis e implantación de mejoras para el Laboratorio de Telemática.

Autor: Daniel Masero Berzal

Tutor: Francisco Javier Muñoz Calle

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

Agradecimientos

He de agradecer a todos aquellos que han hecho posible la realización de este Trabajo de Fin de Grado.

He de agradecer a mis padres y hermano por ser para mí ejemplo a ser, sin su ayuda y esfuerzo no podría haber alcanzado mis metas. Y a mi pareja por ser siempre un apoyo y alentarme a continuar mi camino por duro que fuera este.

Daniel Masero Berzal

Sevilla. 2020

Resumen

El trabajo realizado ha consistido en el estudio de la problemática existente en el laboratorio para cada uno de los puntos en los que aborda el trabajo y el posterior desarrollo de soluciones para cada uno de ellos. Entre la solución aportada se encuentra desde procedimientos para realizar restauraciones de sistema de fichero complejos hasta la generación de modificaciones en el código fuente de Opengnsys.

Para finalizar las soluciones aportadas han sido probadas para ver su viabilidad y su adaptación al problema que se querían abordar.

Abstract

The project has consisted of the study of the problems existing in the laboratory and the development of new solutions for each of these problems. Between the exposed solution, it can be found from procedures for resolving restaurations of a files system to modifications of a Opengnsys source code. At the end, the solutions have been tested to check their viability and their adjustment to the problem

Índice

Agradecimientos vii

Resumen ix

Abstract xi

Índice xiii

Índice de Tablas xviii

Índice de Figuras xx

1 Introducción 11

2 Plan de recuperación y contingencia de pérdidas de datos 12

2.1 Clonación de RAID mediante software *12*

2.1.1 Situación actual y problemática 13

2.1.2 Solución elegida 14

2.1.3 Línea de continuación 26

2.2 Clonación de LVM *27*

2.2.1 Situación actual y problemática 27

2.2.2 Solución elegida 28

2.2.3 Línea de continuación 34

2.3 Copias de seguridad incrementales basado en git *35*

2.3.1 Situación actual y problemática 36

2.3.2 Solución elegida 36

2.3.3 Líneas de continuación 63

2.4 Sincronización de copias de seguridad mediante rsync *64*

2.4.1 Situación actual y problemática 64

2.4.2 Solución elegida 64

2.4.3 Líneas de continuación 72

2.5 Plan de copias de seguridad *72*

2.5.1 Distribución de las unidades de almacenamiento 72

2.5.2 Programa para la realización de las copias de seguridad 73

3 Alta disponibilidad 76

3.1 Configuración de iLO *76*

3.1.1 Situación actual y problemática 76

3.1.2 Configuración de la interfaz de red 77

3.1.3 Actualización de firmware 80

3.1.4 Configuración de puertos y usuarios 85

3.1.5 Líneas de continuación 88

3.2 Gestión de máquinas virtuales QEMU-KVM *89*

3.2.1 Situación actual y problemática 89

3.2.2 Menú de gestión de maquina virtuales QEMU-KVM 89

3.2.3 Configuración de protocolo SPICE en máquinas virtuales QEMU-KVM 93

3.2.4 Líneas de continuación 96

3.3 Clúster de alta disponibilidad *97*

3.3.1 Situación actual y problemática 97

3.3.2 Solución elegida 98
3.3.3 Líneas de continuación 127

4 Despliegue automatizado y gestión remota de equipos 128
4.1 Arranque de máquina virtual de manera automática 128
4.1.1 Situación actual y problemática 128
4.1.2 Solución elegida 128
4.1.3 Líneas de continuación 142
4.2 Modificaciones y procedimientos de Opengnsys 143
4.2.1 Situación actual y problemática 143
4.2.2 Actualización de la versión de Opengnsys a la versión 1.1.1c 144
4.2.3 Corrección en envió de mensajes para realizar Wake On Lan 147
4.2.4 Corrección al realizar la restauración mediante multicast debido a las múltiples subredes del laboratorio 151
4.2.5 Corrección de arranque del cliente ogLive desde cache y configuración del menú de este. 155
4.2.6 Uso de la propiedad de configuración autoexec para realizar la restauración de fichero en el arranque158
4.2.7 Modificación para crear menú para operadores 159
4.2.8 Modificación de la barra superior de la web de gestión de Opengnsys 165
4.2.9 Modificación de la página aula de la web de gestión de Opengnsys 171
4.2.10 Líneas de continuación 176
4.3 Control remoto mediante Guacamole 177
4.3.1 Situación actual y problemática 177
4.3.2 Solución elegida 177
4.3.3 Líneas de continuación 187

5 Gestión de la red interna 189
5.1 Acceso mediante VPN 189
5.1.1 Situación actual y problemática 189
5.1.2 Solución elegida 189
5.1.3 Líneas de continuación 197
5.2 Scripts de gestión de conmutadores 198
5.2.1 Situación actual y problemática 198
5.2.2 Script para la gestión de conmutadores basado en el uso de puerto serie 198
5.2.3 Script para la gestión de conmutadores basado en el uso de SSH 207
5.2.4 Líneas de continuación 210
5.3 Plan de contingencia antes fallo en la red 210
5.3.1 Configuración del conmutador de respaldo 210
5.3.2 Fallo de enrutamiento 212
5.3.3 Fallo tanto enrutamiento como el servicio DHCP 213
5.3.4 Líneas de continuación 213

6 Conclusiones 214

ANEXO A: Clonación de RAID mediante Software 216
Superbloque en RAID software 216
UUIDs de dispositivos perteneciente a un RAID software 217
Script de automatización para la clonación y restauración de RAID software 220
Script el proceso de clonación 220
Script para el proceso de restauración 222

ANEXO B: Clonación de LVM 224
Estructura de fichero generado con vgcfgbackup. 224
Script de automatización para la clonación y restauración de LVM 230
Script para el proceso de clonación 230
Script para el proceso de restauración 231

ANEXO C: Copias de seguridad incrementales basado en git **234**
Conceptos básicos sobre el uso de git *234*
Instalación de la herramienta git-store-meta.pl *235*
Script para la automatización de procedimientos *236*
Servidor 236
Cliente 239
Automatización de los procedimientos mediante el uso de cron *244*

ANEXO D: Sincronización de copias de seguridad mediante rsync **246**
Servidor *246*
Cliente *247*

ANEXO E: Configuración iLO **250**
Activación de la licencia iLO Advanced *250*

ANEXO F: Gestión de máquinas virtuales QEMU-KVM **252**
Proceso de exportación de máquinas virtuales QEMU-KVM. *252*
Proceso de importación de máquinas virtuales QEMU-KVM. *253*
Fichero de configuración configuracionScriptVirsh.cfg *254*

ANEXO G: Clúster de alta Disponibilidad **257**
Configuración de los dispositivos de bloques mediante drbd. *257*
Script para la generación de la regla de redirección de peticiones a un recurso del clúster. *259*

ANEXO H: Arranque de máquina virtual de manera automática **262**
Configuración VNC en máquinas virtuales basadas en VMware *262*
Configuración RDP en máquinas virtuales basadas en VirtualBox *263*
Fichero de configuración reducido para i3. *264*

ANEXO I: Modificaciones y procedimientos de opengnsys **266**
Instalación de cliente ogAgent. *266*
Modificación para crear menú para operadores *266*
Modificación de la barra superior de la web de gestión de Opengnsys *274*
Modificación de la página aula de la web de gestión de Opengnsys *275*
Fichero de configuración configuracion_sincronizacion_guacamole.php *276*
Fichero de configuración configuracion_fichero.php *278*
Fichero de configuración mapeo_particiones_opengnsys.php *279*
Configuración recomendada para gestor_fichero.php *281*

ANEXO J: Control remoto mediante guacamole **284**
Servidor VNC en Linux *284*
Servidor SSH en Linux *286*
Servidor VNC en Windows *287*
Servidor RDP en Windows *290*
Servidor SSH en Windows *291*

ANEXO K: Acceso mediante VPN **294**
Fichero de configuración servidor VPN *294*
Fichero de configuración cliente básico VPN *296*
Servicio para la creación de dispositivos Bridge para OpenVPN *298*
Script detenerBridges.sh 298
Script levantarBridges.sh 298
Script destruirBridge.sh 299
Script levantarBridge.sh 299
Script configuracionBridge.cfg 299
Servicio bridgeOpenvpn.service 300

ANEXO L: Scripts de gestión de conmutadores **302**
Activar servicio ssh *302*

Conmutador Cisco 3750G-24TS 302
Conmutador HP 2620-24 o 2650-24 303
Creación de usuario administrador *304*
Conmutador Cisco 3750G-24TS 304
Conmutador HP 2620-25 o 2650-25 304
Borrado de la configuración total *304*
Conmutador Cisco 3750G-24TS 304
Conmutador HP 2620-25 o 2650-25 306
Referencias **308**

ÍNDICE DE TABLAS

Tabla 1 Copias de seguridad en anfitrión 74
Tabla 2 Copias de seguridad en la máquina virtual 75
Tabla 3 Tipos de UUID en RAID software 220

ÍNDICE DE FIGURAS

Ilustración 2-1 Tabla de partición de /dev/sda 15
Ilustración 2-2 Tabla de partición de /dev/sdb 15
Ilustración 2-3 Escenario propuesto para clonación RAID software 15
Ilustración 2-4 Resultado del comando mdadm sobre /dev/sda1 19
Ilustración 2-5 Resultado del comando mdadm sobre /dev/sda5 20
Ilustración 2-6 Resultado del cálculo de tamaño de sectores 20
Ilustración 2-7 Resultado de la consulta de UUID y etiqueta del RAID /dev/md1 21
Ilustración 2-8 Diagrama de ejemplo de un sistema con LVM 27
Ilustración 2-9 Escenario propuesto para clonación LVM 28
Ilustración 2-10 Resultado del comando blkid sobre /dev/sda1 /dev/sdb1 de LVM 30
Ilustración 2-11 Resultado de la consulta de UUID y etiqueta de partición swap de LVM 31
Ilustración 2-12 Escenario propuesto para las copias de seguridad incrementales con git 37
Ilustración 2-13 Diagrama del estado del repositorio tras crear la rama temporal 48
Ilustración 2-14 Diagrama del estado del repositorio tras forzar el nuevo comienzo del repositorio 49
Ilustración 2-15 Diagrama del estado del repositorio tras borrar la rama temporal 50
Ilustración 2-16 Resultado del comando git log 61
Ilustración 2-17 Escenario propuesto para la sincronización de copias seguridad 65
Ilustración 2-18 Diagrama nueva distribución de las unidades de almacenamiento 72
Ilustración 2-19 Fichero qcwo2 que componen la máquina virtual KVM 73
Ilustración 2-20 Diagrama de servidores del laboratorio 73
Ilustración 3-1 Diagrama de la parte trasera de ProLiant ML310e Gen8 v2 77
Ilustración 3-2 Pantalla de arranque de BIOS de ProLiant ML310e Gen8 v2 78
Ilustración 3-3 Menú de configuración la interfaz iLO 78
Ilustración 3-4 Menú de configuración de red de la interfaz iLO 78
Ilustración 3-5 Funciones de red de la interfaz iLO desactivado 79
Ilustración 3-6 Funciones de red de la interfaz iLO activa y compartida con el Sistema Operativo 79
Ilustración 3-7 Funciones de red de la interfaz iLO activa e independiente 79
Ilustración 3-8 Funciones de red de la interfaz iLO activa e independiente con valores asignados 79
Ilustración 3-9 Menú de configuración de red de la interfaz iLO 80
Ilustración 3-10 Funciones de DHCP de la interfaz iLO destivada 80
Ilustración 3-11 Menú de configuración la interfaz iLO antes de salir 80
Ilustración 3-12 Página de descarga firmware para iLO 81
Ilustración 3-13 Página de descarga firmware concreto para iLO en formato Windows 81
Ilustración 3-14 Menú para la descompresión del firmware de iLO en Windows 82

Ilustración 3-15 Ubicación del fichero firmware de iLO en Windows 82
Ilustración 3-16 Página de descarga firmware concreto para iLO en formato Linux 82
Ilustración 3-17 Ubicación de la página de firmware en el menú lateral de la web de iLO 83
Ilustración 3-18 Página de actualización de firmware de iLO 83
Ilustración 3-19 Popup de confirmación actualización de firmware de iLO 84
Ilustración 3-20 Proceso de subida del firmware de iLO 84
Ilustración 3-21 Proceso de avance subida del firmware de iLO 84
Ilustración 3-22 Inicio actualización del firmware de iLO 84
Ilustración 3-23 Progreso actualización del firmware de iLO 85
Ilustración 3-24 Fin actualización del firmware de iLO 85
Ilustración 3-25 Ubicación de la página la configuración de puertos en el menú lateral de la web de iLO 85
Ilustración 3-26 Página la configuración de puerto de iLO 86
Ilustración 3-27 Popup confirmación para aplicar cambios en la configuración de puertos de iLO 86
Ilustración 3-28 Ubicación de la página la configuración de usuarios en el menú lateral de la web de iLO 87
Ilustración 3-29 Página de usuarios de iLO. 87
Ilustración 3-30 Creación o actualización de usuarios en iLO 88
Ilustración 3-31 Popup de confirmación de eliminación de usuario de iLO 88
Ilustración 3-32 Escenario para las máquinas virtuales 89
Ilustración 3-33 Menú inicial para administración de máquinas virtuales 90
Ilustración 3-34 Menú de estado de las máquinas virtuales 91
Ilustración 3-35 Selección de máquinas virtuales 91
Ilustración 3-36 Menú de gestión de una máquina virtual concreta 92
Ilustración 3-37 Cambio del nombre de una máquina virtual 92
Ilustración 3-38 Menú de importación y exportación de máquinas virtuales 93
Ilustración 3-39 Distribución actual del laboratorio 97
Ilustración 3-40 Distribución de almacenamiento en cada nodo del clúster. 98
Ilustración 3-41 Distribución del clúster en la solución 98
Ilustración 3-42 Resultado comprobación estado sincronización drbd 108
Ilustración 3-43 Distribución inicial de los recursos en el clúster. 116
Ilustración 4-1 Esquema de arquitectura de red simplificado. 143
Ilustración 4-2 Menú de creación de usuario de tipo operador 161
Ilustración 4-3 Unidad administrativa en la cual se encuentra un operador asignado 162
Ilustración 4-4 Información de menú que contiene los diferentes tipos de elementos 162
Ilustración 4-5 Gestión de un menú con los diferentes tipos de elementos 163
Ilustración 4-6 Menú para usuario de tipo administrador 164
Ilustración 4-7 Menú para usuario de tipo operador 164
Ilustración 4-8 Opciones Avanzadas desplegada 170
Ilustración 4-9 Opciones Avanzadas replegada 170

Ilustración 4-10 Página de sincronización entre Opengnsys y Guacamole 170
Ilustración 4-11 Página de gestor de ficheros 170
Ilustración 4-12 Página de conmutador 171
Ilustración 4-13 Aula con operaciones ocultas 175
Ilustración 4-14 Aula con operaciones seleccionadas 175
Ilustración 4-15 Pagina de selección de archivos a enviar 176
Ilustración 4-16 Página para introducir el comando a enviar 176
Ilustración 5-1 Esquema de enrutamiento simplificado del laboratorio de Telemática 189
Ilustración 5-2 Esquema de conexión del conmutador de respaldo 211
Ilustración A-0-1 Diagrama ejemplo de ubicación del superblock version 1.1 217
Ilustración A-0-2 Esquema de UUID en un RAID software 217
Ilustración A-0-3 Resultado de consulta UUID de /dev/sda, /dev/sdb y /dev/md0 218
Ilustración A-0-4 Resultado consulta mdadm -E /dev/sda /dev/sdb 218
Ilustración A-0-5 Resultado consulta mdadm -D /dev/md0 219
Ilustración A-0-6 Escenario propuesto para clonación RAID software 220
Ilustración B-0-1 Esquema de UUID en un LVM 224
Ilustración B-0-2 Resultado de consulta UUID de volúmenes lógicos 224
Ilustración B-0-3 Resultado de consulta UUID de grupo de volúmenes 224
Ilustración B-0-4 Resultado de consulta UUID de volúmenes físicos 225
Ilustración B-0-5 Escenario propuesto para clonación LVM 230
Ilustración C-0-1 Diagrama de zonas de un repositorio de git 234
Ilustración C-0-2 Escenario propuesto para las copias de seguridad incrementales con git 236
Ilustración E-0-1 Ubicación de la página de la licencia en el menú lateral de la web de iLO 250
Ilustración E-0-2 Página de activación de licencia iLO 251
Ilustración E-0-3 Popup de confirmación activación de licencia iLO 251
Ilustración H-0-1 Configuración VNC en VMware 262
Ilustración H-0-2 Configuración RDP en VirtualBox 263
Ilustración J-0-1Selección de la instalación del servidor ThingVNC. 287
Ilustración J-0-2 Selección de ThingVNC como servicio que se autoinicie 288
Ilustración J-0-3 Configuraccion de las contraseñas del servidor ThingVNC 288
Ilustración J-0-4 Parámetros de configuración de ThingVNC pestaña de servidor. 289
Ilustración J-0-5 Parámetros de configuración de ThingVNC pestaña de administración 289
Ilustración J-0-6 Configuración grafica de RDP en Windows 290
Ilustración J-0-7 Configuración grafica de RDPWrapper en Windows 291

1 INTRODUCCIÓN

En este capítulo se explicará los las diferentes mejoras que se han abordado en la realización de este trabajo de fin de grado.

Antes de comenzar es necesario detallara estado actual del laboratorio de Telemática para poder comprender mejor la motivación de las mejoras que se han propuesto.

Actualmente el laboratorio posee una distribución de los equipos existen en dos subredes separadas y la comunicación entre estas subredes y con el exterior es realizar por uno de los servidores existente el cual realiza las funciones encaminador y de firewall.

Dentro del laboratorio hay dos servidores ProLiant ML310e Gen8 v2, uno de ellos siempre está activo y es quien se encarga de proveer los servicios necesarios al laboratorio y realizar las funciones de firewall y enrutador. El otro servidor es una copia con direcciones IP distintas y permanece pausado a la espera de ser usado únicamente cuando se detecte que el servidor activo ha fallado.

En los servidores se encuentra instalado un sistema operativo sobre un conjunto de disco en RAID 1, cuya intención es tener un respaldo ante errores y mejorar la velocidad de lectura, cuya única función es ser el anfitrión de una máquina virtual basada en KVM que será la encargada proveer los servicios necesarios y realizar las funciones de red antes mencionada.

Para finalizar los equipos del laboratorio son administrado mediante el uso de Opengnsys, con el cual se realizan la clonación y restauración de los sistemas operativos usado en estos equipos además de ser el encargado de realizar algunas configuraciones y comprobar el estado de los equipos cuando son usados.

Los objetivos principales de este trabajo son:

- Creación de un sistema de copias de seguridad y métodos recuperación para los datos almacenados en los servidores del laboratorio.
- Creación de un sistema de alta disponibilidad para las máquinas virtuales usadas en los servidores del laboratorio.
- Creación de un plan de contingencia ante fallo de red.
- Corrección y mejora en control y estado de los equipos administrado con Opengnsys.
- Permitir el acceso remoto a la red interna del laboratorio y el control remoto de sus equipos.

2 PLAN DE RECUPERACIÓN Y CONTINGENCIA DE PÉRDIDAS DE DATOS

En este capítulo se abordará una serie de procedimientos que permitirán la realización de copias de seguridad de los datos que son almacenados y usado en el Laboratorio de telemática. Tambien se detallará un plan de copias de seguridad a realizar para mantener las copias actualizada y evitar la pérdida de información debido a fallo de un servidor o eliminación de parte de las copias en un servidor.

2.1 Clonación de RAID mediante software

En esta sección se detallará un procedimiento por el cual poder clonar sistemas de archivos creados sobre unidades de almacenamiento RAID, ya creadas y mantenidas mediante software.

Antes de avanzar se explicará brevemente a qué se refiere cuando se usa la terminología RAID. Un RAID es un grupo de dispositivos de almacenamiento o particiones que se configuran para que trabajen de forma conjunta de manera que el sistema operativo que lo utiliza lo ve como un único dispositivo. Según la configuración, lo que se denomina nivel de RAID[1] puede proporcionar diferentes ventajas: una mayor velocidad de escritura, una mayor velocidad de lectura, tolerancia ante errores o una combinación de ambas. Pero también tiene desventajas, ya que se pierde capacidad de almacenamiento efectiva y velocidad de lectura/escritura o ambas respecto al uso del dispositivo de forma separada.

En el

ANEXO A: Clonación de RAID mediante Software se detallará que es un superbloque y como se relacionan los dispositivos y el RAID software que componen mediante sus UUID. Estos conceptos son útiles para comprender mejor el proceso de clonación y restauración.

Debido a la ventaja que proporcionan estos sistemas, tanto en redundancia de los datos como en velocidad de lectura, se usan en el laboratorio de Telemática como base para después realizar las particiones necesarias para los sistemas operativos y las aplicaciones usadas.

2.1.1 Situación actual y problemática

Actualmente el laboratorio de Telemática dispone de dos servidores ProLiant ML310e Gen8 v2 que poseen un hardware HP Dynamic Smart Array B120i/ZM, que es una controladora RAID. Esta permite la administración de RAIDs mediante hardware. El uso de esta tecnología proporcionará un mayor rendimiento, ya que se usaría hardware dedicado a la sincronización de los datos entre los discos que forman el RAID y el cálculo de datos necesarios según la configuración del RAID elegida. Así se conseguiría descargar la CPU. Además, al tratarse de hardware dedicado, es más fiable ante posibles fallos.

El problema que presenta esta solución, y por lo cual no se usa actualmente en el laboratorio para la administración de RAIDs, es que los drivers que son necesarios para que los sistemas operativos detecten el RAID como una sola unidad de almacenamiento son propietario de HP y únicamente están disponibles para los siguientes sistemas operativos[2]:

- Microsoft Windows 2012 Server (64 bit)
- Microsoft Windows 2008 Server R2 (64 & 32 bit)
- Red Hat Enterprise Linux 5 (64 & 32 bit)
- Red Hat Enterprise Linux 6
- SUSE Linux Enterprise Server 10 (64 & 32 bit)
- SUSE Linux Enterprise Server 11 (64 & 32 bit)
- VMware ESX 4.1 u2
- VMware ESXi 4.1 u2
- VMware ESXi 5.0, 5.0 u1, 5.0 u2, ESXi 5.1

Todos estos sistemas operativos son de pago y se prefiere el uso de alternativas basadas en software de código abierto. Al no poseer un controlador compatible para el sistema operativo, que en el laboratorio es Debian, no se detectará las unidades de almacenamiento como si de una única unidad RAID se tratara, sino que lo verá como componentes independientes de este.

Como alternativa se hace uso del software mdadm[3] que permite la administración completa de almacenamientos RAID software. Este es soportado por la gran mayoría de las distribuciones Linux y permite que el gestor de arranque sea instalado sobre particiones ya creadas que se encuentren en un RAID.

Pero al hacer uso de esta alternativa se hace más complejo el método para realizar copias de seguridad de las unidades de almacenamientos de forma completa. El problema en cuestión es que actualmente no existe ninguna herramienta o procedimiento eficiente basado en código abierto que permita la creación de estas copias de seguridad y su posterior restauración.

Una posible solución sería el uso de la herramienta dd para realizar la copia de seguridad. Sin embargo, esto implicaría generar una copia cuyo tamaño sería la suma del tamaño de cada unidad almacenamiento que compone nuestro sistema, además, se tendría como limitación que esta herramienta no permite el volcado de la información sobre discos de un tamaño inferior.

Por ejemplo, si hacemos uso de esta herramienta sobre un sistema donde disponemos de 2 discos duros de 1Tb

de capacidad de almacenamiento que forman un RAID 1 tendríamos que generar la copia de ambos discos teniendo como resultado una copia de seguridad de 2Tb.

2.1.2 Solución elegida

A continuación, se detallará el procedimiento que se ha de seguir para realizar una copia de seguridad completa de un sistema que posee varios RAIDs y su posterior uso para realizar la restauración de los datos a partir de esta.

En el

ANEXO A: Clonación de RAID mediante Software se expondrá los scripts necesarios para poder realizar estos procesos de forma automatizada.

Se empezará describiendo la actual distribución de unidades de almacenamientos de los servidores del laboratorio de Telemática. Esta disposición se usará como escenario de ejemplo para explicar los pasos a seguir:

- Disco /dev/sda cuya capacidad es de 931,5 GiB y sus particiones

Ilustración 2-1 Tabla de partición de /dev/sda

- Disco /dev/sdb cuya capacidad es de 931,5 GiB y sus particiones

Ilustración 2-2 Tabla de partición de /dev/sdb

- Se ha creado un RAID-1 /dev/md0 usando para ello las particiones:
  - /dev/sda1
  - /dev/sdb1

Este RAID se ha formateado con el sistema de archivos ext4 donde se ha instalado el sistema operativo Debian 8.

- Se ha creado un RAID-1 /dev/md1 usando para ello las particiones:
  - /dev/sda5
  - /dev/sdb5

Este RAID se ha formateado para ser usado como zona de intercambio del sistema Debian.

A continuación, se muestra un diagrama con la distribución de particiones y RAID software.

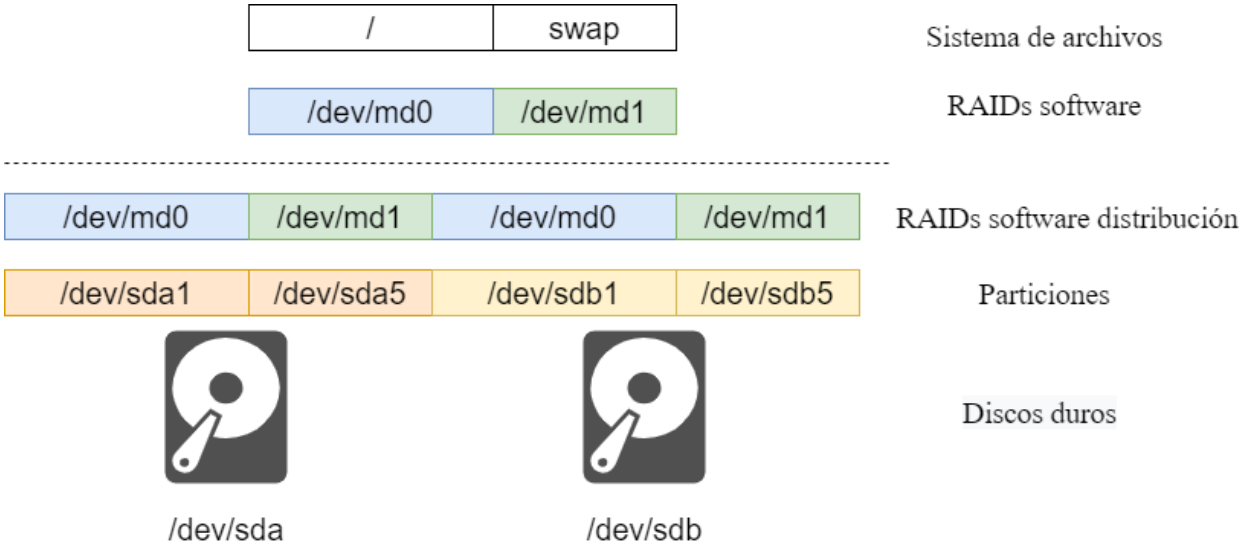

Ilustración 2-3 Escenario propuesto para clonación RAID software

Por último, para llevar a cabo tanto la clonación como su posterior restauración se hará uso de un USB de arranque con la distribución SystemRescueCd cuya versión debe ser 6.0.0 o superior, ya que incluye todas las herramientas necesarias para poder realizar los procesos de fsarchiver[4], mdadm y dd[5].

2.1.2.1 Proceso de clonación

A continuación, se detallará el proceso a seguir para realizar la clonación completa del escenario propuesto.

En

ANEXO A: Clonación de RAID mediante Software se proporciona un script que realiza automáticamente todos los pasos que se detallan.

Los pasos que se siguen son los siguientes:

1. Clonar el contenido de la partición con datos: en este ejemplo será /dev/md0. Para ello se utiliza la herramienta fsarchiver. Entre otras características de esta herramienta se encuentran la de restaurar posteriormente sobre discos con un tamaño menor que el original, clonar sistemas de archivos ext4 y ejecutar la compresión del archivo de backup de forma automática.

```
fsarchiver -j2 -v savefs md0_backup /dev/md0
```

Los argumentos usados son:

-j2	Indica el número de hilo que se usará en el proceso clonación. Permite que el proceso se realice de manera más rápida si se dispone de un procesador con varios núcleos. El número 2 indica que se utilizan 2 núcleos.
-v	Activa el modo verboso para poder ver el progreso de la clonación.
savefs	Indica que se va a realizar la clonación de un sistema de archivos.
md0_backup	Indica el archivo sobre el que se quiere volcar el contenido de la clonación.
/dev/md0	Indica el dispositivo que se quiere clonar.

2. Clonar el registro de arranque maestro (MBR) de los discos /dev/sda y /dev/sdb. Para ello se hará uso de la herramienta dd que permite clonar exactamente un número determinado de bytes. Esta propiedad es muy útil al conocer que el MBR se encuentra alojado en los primeros 512B de los discos.

```
dd if=/dev/sda of=backup_sda_MBR.dd bs=512 count=1
dd if=/dev/sdb of=backup_sdb_MBR.dd bs=512 count=1
```

Los argumentos usados son:

if=/dev/sdX	Indica el origen de donde se leerán los bytes.
of=backup_sda_MBR.dd	Indica el destino sobre el que se volcarán los bytes leídos.
bs=512	Indica el tamaño del bloque de los datos que leerán del origen y que se escribirán en el destino de una vez. Si no se pone ningún sufijo, se estará indicando en bytes. El valor en 512 por ser el tamaño del MBR.
count=1	Indica el número de bloques que se copiarán.

3. Clonar la tabla de particiones de los discos /dev/sda y /dev/sdb. Para ello se utilizará la herramienta sfdisk[6] que permite realizar la clonación de la tabla de partición a un archivo de forma sencilla únicamente mediante un comando.

```
sfdisk -d /dev/sda > backup_sda.sfdisk
sfdisk -d /dev/sdb > backup_sdb.sfdisk
```

Los argumentos usados son:

-d	Indica que se imprima por la salida la tabla de particiones con el formato adecuado para usarse como entrada del comando sfdisk y así recrear la tabla de particiones.
/dev/sdX	Indica el dispositivo del cual se quiere clonar la tabla de particiones.

4. Calcular la ubicación de los superbloques que contienen la información de los RAIDs /dev/md0 y /dev/md1. Para ello se hará uso de la herramienta mdadm con la cual se administran los RAIDs software. En el

5. ANEXO A: Clonación de RAID mediante Software se detallará que es el super bloque y porque es necesario clonarlo.

Para conocer la ubicación del superbloque, primero se debe escoger a uno de los dispositivos que compongan el RAID. Para el RAID /dev/md0 se cogerá la partición /dev/sda1 y para /dev/md1, la partición /dev/sda5.

```
mdadm -E /dev/sda1
mdadm -E /dev/sda5
```

El argumento usado es:

-E	Indica que se quiere ver el contenido de los metadatos
/dev/sdX	Indica el dispositivo del cual se quiere obtener los metadatos.

Tras ejecutar el primero de los comandos se obtendrá un resultado como el siguiente:

```
root@lt-servidor2:~# mdadm -E /dev/sda1
/dev/sda1:
          Magic : a92b4efc
        Version : 1.2
    Feature Map : 0x0
     Array UUID : 2adb0988:d29e7a10:c090edd8:938a21a0
           Name : servidor2:1
  Creation Time : Sun May 31 16:31:18 2020
     Raid Level : raid1
   Raid Devices : 2

 Avail Dev Size : 39028736 (18.61 GiB 19.98 GB)
     Array Size : 19514368 (18.61 GiB 19.98 GB)
    Data Offset : 32768 sectors
   Super Offset : 8 sectors
   Unused Space : before=32680 sectors, after=0 sectors
          State : clean
    Device UUID : d89b18f7:051263a7:b5ad48e2:7dabfade

    Update Time : Fri Feb 15 16:31:36 2019
  Bad Block Log : 512 entries available at offset 72 sectors
       Checksum : c37c1ba5 - correct
         Events : 84


   Device Role : Active device 0
   Array State : AA ('A' == active, '.' == missing, 'R' == replacing)
```

Ilustración 2-4 Resultado del comando mdadm sobre /dev/sda1

De este resultado se debe guardar el valor del camp Super Offset que indica a que distancia desde el comienzo de la partición /dev/sda1 se encuentra el superbloque. Este será igual para la partición /dev/sdb1. El valor de offset es de 8 sectores.

Tras ejecutar el segundo comando se obtendrá un resultado como el siguiente:

```
root@lt-servidor2:~# mdadm -E /dev/sda5
/dev/sda5:
          Magic : a92b4efc
        Version : 1.2
    Feature Map : 0x1
     Array UUID : a9cb7e4a:6808a3e6:26123963:62c72404
           Name : servidor2:2
  Creation Time : Sun May 31 16:31:33 2020
     Raid Level : raid1
   Raid Devices : 2

 Avail Dev Size : 1914198016 (912.76 GiB 980.07 GB)
     Array Size : 957099008 (912.76 GiB 980.07 GB)
    Data Offset : 262144 sectors
   Super Offset : 8 sectors
   Unused Space : before=262056 sectors, after=0 sectors
          State : active
    Device UUID : fd0fc41d:ed84b410:ce7ef93a:d300c560

Internal Bitmap : 8 sectors from superblock
    Update Time : Fri Feb 15 17:39:46 2019
  Bad Block Log : 512 entries available at offset 72 sectors
       Checksum : 946e975e - correct
         Events : 9783


   Device Role : Active device 0
   Array State : AA ('A' == active, '.' == missing, 'R' == replacing)
```

Ilustración 2-5 Resultado del comando mdadm sobre /dev/sda5

Se deberá guardar el mismo valor para /dev/sda5 y /dev/sdb5. El valor de offset es de 8 sectores.

En este ejemplo, los valores coinciden, pero no tiene por qué ser así. En este caso ocurre porque la versión de los superbloques es para ambos RAIDs la v1.2.

6. Calcular el tamaño de los sectores para cada dispositivo que compone los RAIDs.

```
cat /sys/block/sda/queue/hw_sector_size
cat /sys/block/sdb/queue/hw_sector_size
```

El resultado obtenido es el siguiente:

```
root@lt-servidor2:~# cat /sys/block/sda/queue/hw_sector_size
512
root@lt-servidor2:~# cat /sys/block/sdb/queue/hw_sector_size
512
```

Ilustración 2-6 Resultado del cálculo de tamaño de sectores

Esto indica que tanto el dispositivo /dev/sda y /dev/sba tienen sectores de 512B.

7. Clonar los superbloques de todos los dispositivos que componen los diferentes RAIDs. Para ello se usará de nuevo la herramienta dd que permite indicar el punto concreto desde donde comenzar la lectura de información para la clonación.

```
dd if=/dev/sda1 of=sda1_superblock.dd count=1 bs=512 skip=8
dd if=/dev/sdb1 of=sdb1_superblock.dd count=1 bs=512 skip=8
dd if=/dev/sda5 of=sda5_superblock.dd count=1 bs=512 skip=8
dd if=/dev/sdb5 of=sdb5_superblock.dd count=1 bs=512 skip=8
```

Los argumentos usados son:

if=/dev/sdX	Indica el origen desde donde se leerán los bytes.
of= backup_sda_MBR.dd	Indica el destino sobre el que se volcarán los bytes leídos.
bs=512	Indica el tamaño del bloque de datos que se leerá del origen y se escribirá en el destino de una vez. Por defecto, si no se pone ningún sufijo, se estará indicando en bytes. En este caso serán 512 al ser el tamaño de los sectores de los discos.
count=1	Indica el número de bloques que se copiarán. Este será calculado en función del número de dispositivos que compongan el RAID.
skip=8	Indica cuántos bloques se saltarán antes de comenzar la lectura de los datos en el origen. Este valor será el que indique el campo Super Offset calculado en el paso 4.

En este punto será necesario calcular el tamaño de los bloques y el número de bloque a copiar. Para ello hay que aplicar la siguiente fórmula:

$$T_s = 260 + 2 \times N_d$$

Donde T_s será el número de bytes que ocupa el superbloque y N_d el número de dispositivos que forman el RAID.

Esto nos indica cuántos bloques del tamaño de los sectores serán necesario clonar. No debe importar si se clona más información de la que forma parte estrictamente del superbloque porque no influirá en el procedimiento de restauración.

8. Para finalizar, al poseer una partición swap en /dev/md1, se necesita guardar su UUID y su etiqueta, si la tuviera, para poder recrearla correctamente. Para ello, se utilizará la herramienta blkid[7] que indica esta información.

```
blkid /dev/md1
```

El argumento usado es:

/dev/sdX	Indica el dispositivo del cual se quiere obtener su UUID y su etiqueta

El resultado del comando anterior será el siguiente:

```
root@lt-servidor2:~# blkid /dev/md1
/dev/md1: LABEL="servidor2-swap" UUID="ee04f9c-a191-42de-814c-1fb67ba0cbed"
 TYPE="swap"
```

Ilustración 2-7 Resultado de la consulta de UUID y etiqueta del RAID /dev/md1

De donde debemos guardar el valor del campo LABEL y del campo UUID.

2.1.2.2 Proceso de restauración

A continuación, se explicará el proceso que se sigue para realizar la restauración completa del escenario propuesto a partir de los ficheros y datos obtenidos en el proceso de clonación.

En el

ANEXO A: Clonación de RAID mediante Software se proporciona un script que realiza automáticamente todos los pasos que se detallarán.

Los pasos que seguir son los siguientes:

1. Restaurar el registro de arranque maestro (MBR) de los discos /dev/sda y /dev/sdb. Se ejecutará la herramienta dd con la que se generó la copia y que permite el proceso inverso.

```
dd of=/dev/sda if=backup_sda_MBR.dd bs=512 count=1
dd of=/dev/sdb if=backup_sdb_MBR.dd bs=512 count=1
```

Los argumentos usados son:

if=backup_sda_MBR.dd	Indica el origen desde donde se leerán los bytes.
of=/dev/sdX	Indica el destino sobre el que se volcarán los bytes leídos.
bs=512	Indica el tamaño del bloque de los datos que se leerán del origen y que se escribirán en el destino de una vez. Por defecto, si no se pone ningún sufijo, se estará indicando en bytes. El valor es 512 por ser el tamaño del MBR.
count=1	Indica el número de bloques que se copiarán.

2. Restaurar la tabla de particiones de los discos /dev/sda y /dev/sdb. Se hará uso de la herramienta sfdisk que fue la que se utilizó para la creación del archivo con la tabla de particiones.

```
sfdisk /dev/sda < backup_sda.sfdisk
sfdisk /dev/sdb < backup_sdb.sfdisk
```

Los argumentos usados son:

/dev/sdX	Indica el dispositivo del cual se quiere clonar la tabla de particiones

Como entrada de la herramienta se usa el fichero generado en el proceso de clonación de la tabla de particiones. Este archivo debe poseer un formato adecuado.

3. Restaurar los superbloques de todos los dispositivos que componen los diferentes RAIDs. Se usará de nuevo el comando dd que permite indicar el punto desde donde comenzar la escritura.

```
dd of=/dev/sda1 if=sda1_superblock.dd bs=512 seek=8
dd of=/dev/sdb1 if=sdb1_superblock.dd bs=512 seek=8
dd of=/dev/sda5 if=sda5_superblock.dd bs=512 seek=8
dd of=/dev/sdb5 if=sdb5_superblock.dd bs=512 seek=8
```

Los argumentos usados son:

if= backup_sda_MBR.dd	Indica el origen desde donde se leerán los bytes.
of=/dev/sdX	Indica el destino sobre el que se volcarán los bytes leídos.

bs=512	Indica el tamaño del bloque de datos que se leerán del origen y que se escribirán en el destino de una vez. Por defecto, si no se pone ningún sufijo, se estará indicando en bytes. Este valor se deberá haber obtenido en el paso 5 del proceso de clonación.
seek=8	Indica cuántos bloques se saltarán antes de comenzar la escritura de los datos en el destino. Este valor se deberá haber obtenido en el paso 4 del proceso de clonación.

4. Se deberá detener todos los RAIDs por si se hubieran iniciado automáticamente. Para conseguirlo se usará la herramienta mdadm con la cual se administran los RAIDs software.

```
mdadm -S --scan
```

Los argumentos usados son:

-S	Indica que se desactive el RAID administrado por mdadm.
--scan	Indica que se busquen todos los RAIDs administrados por mdadm en el sistema. Esto provocará que se detenga los RAIDs software.

5. Recrear los RAIDs para poder trabajar más fácilmente con ellos. Se usará la herramienta mdadm de nuevo.

```
mdadm -A --run --update=resync /dev/md0 /dev/sda1 /dev/sdb1
mdadm -A --run --update=resync /dev/md1 /dev/sda5 /dev/sdb5
```

Los argumentos usados son:

-A	Indica que se quiere montar un RAID a partir de un dispositivo que antes formaba parte de otro.
--run	Indica que se forme el RAID, aunque alguno de sus dispositivos ya esté en otro RAID.
--update=resync	Indica que mientras se rearme el RAID se sincronice los dispositivos. Usándolo se evita la redundancia del RAID-1.
/dev/mdX	Indica el dispositivo RAID que se quiere crear.
/dev/sdaY/ dev/sdbY	Indica los dispositivos que forman el RAID.

6. Restaurar el contenido de la partición con datos. En este ejemplo será /dev/md0. Para ello se utilizará la herramienta fsarchiver de nuevo.

```
fsarchiver -j2 -v restfs md0_backup.fsa id=0,dest=/dev/md0
```

Los argumentos usados son:

-j2	Indica el número de hilo que se usará en el proceso de clonación. Permite que el proceso se realice de manera más rápida si se dispone de un procesador con varios núcleos. El numero 2 indica que se usarán 2 núcleos.
-v	Activa el modo verboso para poder ver el progreso de la clonación.
restfs	Indica que se va a realizar la restauración de un sistema de archivos.
md0_backup.fsa	Indica el archivo que se usará como origen para la restauración.
id=0	Indica que se restaure el primer sistema de archivo que se encuentre en este.
dest=/dev/md0	Indica el destino sobre el que se realizará la restauración.

7. Recrear la partición swap en /dev/md1. Para ello se usará la mkswap[8].

```
 mkswap -U "ee04f9fc-a191-42de-814c-1fb67ba0cdeb" -L "servidor2-swap"
/dev/md1
```

Los argumentos usados son:

-U "ee04f9fc-a191-42de-814c-1fb67ba0cdeb"	Indica el UUID que se le asignará a la partición swap una vez que se cree.
-L "servidor2-swap"	Indica la etiqueta que se le asignará a la partición swap una vez se crea. Si no la tuviera antes, este argumento se podría omitir.
/dev/md1	Indica el dispositivo en el que se creará la partición swap.

8. Para finalizar se debe reinstalar el gestor de arranque grub. Para ello se hará uso de la herramienta grub-install[9] junto con la herramienta mount[10] y mkdir[11].

```
mkdir md0_carpeta
mount /dev/md0 md0_carpeta
grub-install --boot-directory=md0_carpeta/boot/ /dev/sda
grub-install --boot-directory=md0_carpeta/boot/ /dev/sdb
```

El argumento usado para la herramienta mkdir[11] es:

md0_carpeta	El nombre del directorio que se crea.

Los argumentos usados para la herramienta mount[10] son:

/dev/md0	Indica el dispositivo que se monta.

md0_carpeta	Indica el lugar dónde realizar el montaje del dispositivo

Los argumentos usados para la grub-install[9] son:

--boot-directory=md0/boot	Indica el directorio sobre el que instalar el GRUB.
/dev/sdX	Indica el dispositivo en cuyo sector de arranque se instalará el GRUB.

2.1.3 Línea de continuación

Una posible línea de continuación puede ser estudiar como extraer los drivers HP Dynamic Smart Array para un sistema operativo de pago y realizar la instalación sobre otros sistemas operativos. De esta manera el proceso de clonación y restauración se simplificaría, se consumiría menos recursos del sistema y sería más fiable.

2.2 Clonación de LVM

En este apartado se detallará un procedimiento por el cual clonar sistemas de archivos creados sobre unidades de almacenamiento LVM[12].

Antes de avanzar se explicará brevemente a qué se esta haciendo referencia cuando se habla de LVM[12]. Su nombre completo es "gestor de volúmenes lógicos" y consiste en una funcionalidad de Linux que permite administrar los dispositivos y sus particiones de una forma más sencilla. Esta funcionalidad permite crear particiones virtuales que pueden agrupar varios dispositivos de almacenamiento o varias particiones de este elemento o de otro diferente; permite redimensionar estas particiones virtuales sin tener que estar limitado por el espacio contiguo disponible, sino únicamente por espacio total disponible.

El funcionamiento de LVM se basa principalmente en tres conceptos:

- Volúmenes físicos (PV, Physical volume). Hace referencia a los dispositivos de almacenamiento del sistema que proporcionará el espacio del almacenamiento para las particiones virtuales. Un ejemplo de esto sería un raid software, una partición del disco duro o el disco duro completo.
- Grupo de volúmenes (GV, Volume group). Consiste en un conjunto de volúmenes físicos que sirve como base para realizar la partición.
- Volúmenes lógicos (LV, Logical volume). Hace referencia a la partición virtual propiamente dicha. Se encuentra ubicada en un grupo de volúmenes. Esta puede ocupar tanto espacio como el grupo de volúmenes no tenga ya asignado a otro volumen lógico y puede decrecer tanto como su sistema de fichero o tipo de partición se lo permita.

A continuación, podemos ver un diagrama en el que se ve un ejemplo del uso de LVM en un sistema.

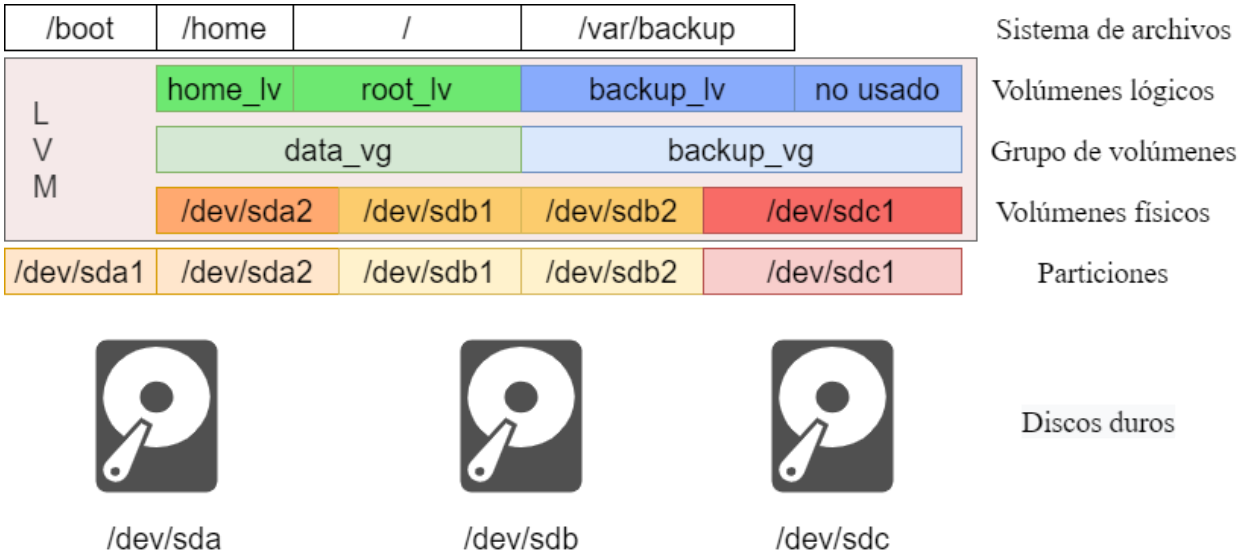

Ilustración 2-8 Diagrama de ejemplo de un sistema con LVM

2.2.1 Situación actual y problemática

El laboratorio de Telemática dispone de dos servidores. En cada uno de ellos hay instalados dos discos SSD de un 1Tb de capacidad en RAID 1 creados mediante software. En estos RAIDs se almacenan las copias de seguridad del laboratorio y toda la información necesaria para el funcionamiento de este.

Recientemente se instalaron dos discos SSD de 2Tb de capacidad por cada servidor en RAID 1. Estos elementos serán destinados principalmente para almacenar las copias de seguridad, aunque se desea tener la posibilidad de utilizar parte de su capacidad de memoria para el funcionamiento del laboratorio por si es necesario en el futuro.

Debido a esto se hará uso de LVM. Esta tecnología permite el redimensionamiento de las particiones de forma sencilla una vez creadas. También es capaz de seguir aumentando la capacidad de almacenamiento añadiendo

nuevos discos duros al grupo de volúmenes para posteriormente aumentar la capacidad del volumen lógico que se quiera.

2.2.2 Solución elegida

A continuación, se detallará el procedimiento que se ha de seguir para realizar una copia de seguridad completa de un sistema que posee varios volúmenes lógicos y su posterior uso para realizar la restauración de los datos a partir de esta.

En el ANEXO B: Clonación de LVM se expondrá los scripts necesarios para poder realizar estos procesos de forma automática.

En esta subsección se describe un escenario simplificado que no se adapta totalmente al existente. El motivo es que no se parte de la existencia de RAID software como dispositivos que se conviertan en volúmenes lógicos.

En caso de poseer un RAID software habría que clonarlo o restaurarlo antes de realizar el procedimiento que se indicará luego y con lo visto en la sección Clonación de RAID mediante software.

El escenario que se utilizará para explicar el procedimiento será un sistema que posee dos discos duros cuyo particionamiento es el siguiente:

- El disco /dev/sda tiene dos particiones: la primera sería /dev/sda1, que se usará para instalar el gestor de arranque, ya que no se puede encontrar dentro de un dispositivo LVM, y la segunda, un volumen físico.
- El disco /dev/sdb solo dispone de una partición que será un volumen físico.

En el escenario se tiene un grupo de volúmenes llamados volumenDatos. Este conjunto está compuesto por los volúmenes físicos /dev/sda2 y /dev/sdb1. De este grupo se crean dos volúmenes lógicos:

- El volumen lógico llamado root, donde se alojará el directorio raíz del sistema operativo.
- El volumen lógico llamada swap, que será usado como swap del sistema operativo.

En la imagen se puede observar un diagrama con el escenario propuesto.

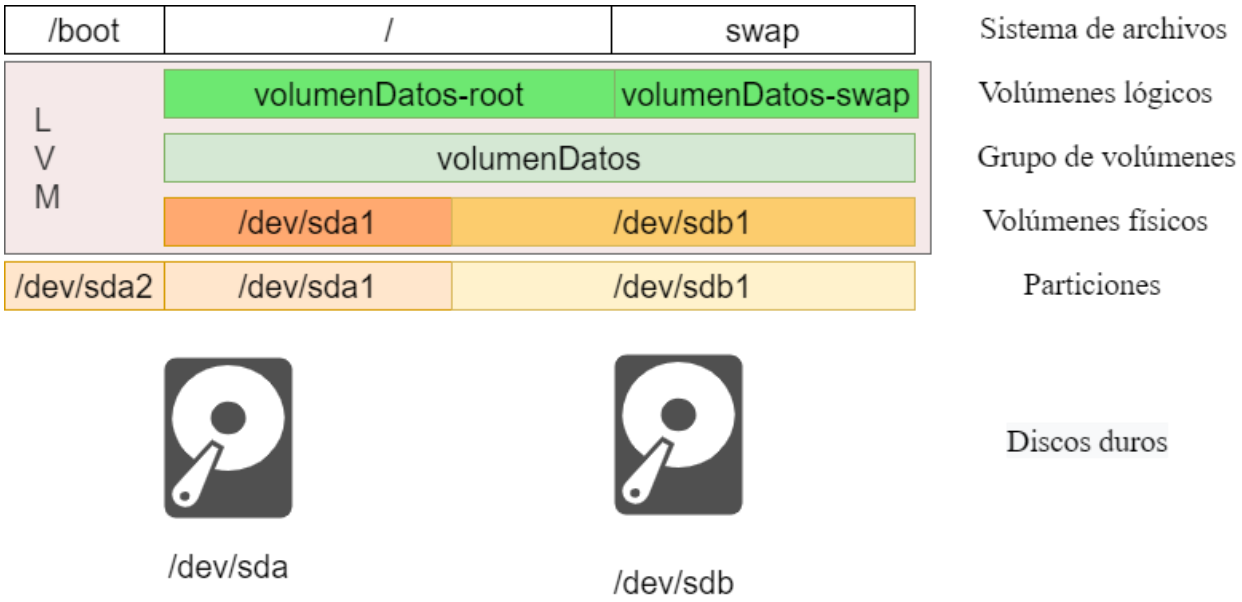

Ilustración 2-9 Escenario propuesto para clonación LVM

Por último, para llevar a cabo tanto la clonación como su posterior restauración, se hará uso de un USB de arranque con la distribución SystemRescueCd cuya versión debe ser 6.0.0 o superior. Esta versión incluye todas las herramientas necesarias para poder realizar los procesos de fsarchiver, vgcfgrestore[13], vgcfgbackup[14], vgchange[15], pvcreate[16] y dd.

2.2.2.1 Proceso de clonación

En el ANEXO B: Clonación de LVM se proporciona un script que realiza automáticamente todos los pasos que se detallan.

Los pasos que se deben seguir son los siguientes:

1. Clonar el contenido de la partición con datos. En este ejemplo serán /dev/mapper/volumenDatos-root y /dev/sda1. Para ello se ejecuta la herramienta fsarchiver. Entre otras características de esta herramienta se encuentran la de restaurar posteriormente sobre discos con un tamaño menor que el original, clonar sistemas de archivos ext4 y realizar la compresión del archivo de backup de forma automática.

```
fsarchiver -j2 -v savefs root_backup /dev/mapper/volumenDatos-root
fsarchiver -j2 -v savefs boot_backup /dev/sda2
```

Los argumentos usados son:

-j2	Indica el número de hilo que se usará en el proceso clonación. Permite que el proceso se realice más rápido si se dispone de un procesador con varios núcleos. El número 2 indica que se usan 2 núcleos.
-v	Activa el modo verboso para poder ver el progreso de la clonación.
savefs	Indica que se va a realizar la clonación de un sistema de archivos.
X_backup	Indica el archivo sobre el que se quiere volcar el contenido de la clonación.
/dev/X	Indica el dispositivo que se quiere clonar.

2. Clonar el registro de arranque maestro (MBR) de los discos /dev/sda y /dev/sdb. Para ello se hará uso de la herramienta dd. Esta nos permite clonar exactamente un número determinado de bytes. Esta propiedad es muy útil al conocerse que el MBR se encuentra alojado en los primeros 512B de los discos.

```
dd if=/dev/sda of=backup_sda_MBR.dd bs=512 count=1
dd if=/dev/sdb of=backup_sdb_MBR.dd bs=512 count=1
```

Los argumentos usados son:

if=/dev/sdX	Indica el origen desde donde se leerán los bytes.
of=backup_sda_MBR.dd	Indica el destino sobre el que se volcarán los bytes leídos.
bs=512	Indica el tamaño del bloque de los datos que se leerán del origen y que se escribirán en el destino de una vez. Si no se pone ningún sufijo, se estará indicando en bytes. El valor en 512 por ser el tamaño del MBR.
count=1	Indica el número de bloques que se copiarán.

3. Clonar la tabla de particiones de los discos /dev/sda y /dev/sdb. Para ello se utilizará la herramienta sfdisk que se encarga de realizar la clonación de la tabla de partición y guardarla en un archivo de forma sencilla, únicamente mediante un comando.

```
sfdisk -d /dev/sda > backup_sda.sfdisk
sfdisk -d /dev/sdb > backup_sdb.sfdisk
```

Los argumentos usados son:

-d	Con esta opción se imprime en la salida la tabla de particiones con el formato adecuado que se usará como entrada del comando sfdisk[6] y así recrear la tabla de particiones.
/dev/sdX	Indica el dispositivo del que se quiere clonar la tabla de particiones.

4. Volcar la configuración LVM del grupo de volúmenes volumenDatos. Para ello se emplea la herramienta vgcfgbackup que permite el volcado de la configuración en un fichero. Este archivo será usado posteriormente como base para su restauración.

```
vgcfgbackup -f volumenDatosConfig volumenDatos
```

Los argumentos usados son:

-f volumenDatosConfig	Indica el fichero donde se volcará la configuración LVM del grupo de volúmenes.
volumenDatos	Indica el grupo de volúmenes de donde se quiere volcar la configuración.

5. Se obtienen los UUIDs de los volúmenes físicos que componen el grupo de volúmenes volumenDatos, esto es necesario para poder reconstruir posteriormente la configuración LVM. Para ello se usará la herramienta blkid que indica esta información.

```
blkid /dev/sda1 /dev/sdb1
```

El argumento usado es:

/dev/sdX1	Indica el dispositivo cuyo UUID se quiere obtener.

El resultado del comando anterior será el siguiente:

```
[root@lt-servidor2 ~]# blkid /dev/sda1 /dev/sdb1
/dev/sda1: UUID="Ds1dCd-Ow6m-nnyY-2jcI-MDtq-Rw1n-BZcpfB" TYPE="LVM2_member"
 PARTUUID="000567d5-01"
/dev/sdb1: UUID="Ya1SlF-Rmif-8vQd-yfSd-twiM-77su-Z3VKai" TYPE="LVM2_member"
 PARTUUID="0004d58a-01"
```

Ilustración 2-10 Resultado del comando blkid sobre /dev/sda1 /dev/sdb1 de LVM

De donde debemos guardar el valor del campo UUID.

6. Para finalizar, al tener una partición swap en /dev/mapper/volumenDatos-swap, se necesita guardar su UUID y su etiqueta en caso de que la tuviera para poder recrearla correctamente. Para ello, se utilizará la herramienta blkid que indica esta información.

```
blkid /dev/mapper/volumenDatos-swap
```

El argumento usado es:

/dev/mapper/volumenDatos-swap	Indica el dispositivo del que se quiere obtener el UUID y la etiqueta

El resultado del comando anterior será el siguiente:

```
[root@lt-servidor2 ~]# blkid /dev/mapper/volumenDatos-swap
/dev/mapper/volumenDatos-swap: LABEL="servidor2-swap" UUID="86379206-3568-48ab
-bf55-d87e46fd337c" TYPE="swap"
```

Ilustración 2-11 Resultado de la consulta de UUID y etiqueta de partición swap de LVM

De aquí debemos guardar el valor del campo LABEL y del campo UUID.

2.2.2.2 Proceso de restauración

En el ANEXO B: Clonación de LVM se proporcionará un script que realiza automáticamente todos los pasos que se detallan.

Los pasos que se siguen son los siguientes:

9. Restaurar el registro de arranque maestro (MBR) de los discos /dev/sda y /dev/sdb. Se hará uso de la herramienta dd con la que se generó la copia y que permite el proceso inverso.

```
dd of=/dev/sda if=backup_sda_MBR.dd bs=512 count=1
dd of=/dev/sdb if=backup_sdb_MBR.dd bs=512 count=1
```

Los argumentos usados son:

if=backup_sda_MBR.dd	Indica el origen donde se leerán los bytes.
of=/dev/sdX	Indica el destino sobre el que se volcarán los bytes leídos.
bs=512	Indica el tamaño del bloque de los datos que se leerán del origen y que se escribirán en el destino de una vez. Por defecto, si no se pone ningún sufijo, se estará indicando en bytes. El valor es 512 por ser el tamaño del MBR.
count=1	Indica el número de bloques que se copiarán.

10. Restaurar la tabla de particiones de los discos /dev/sda y /dev/sdb. Se hará uso de la herramienta sfdisk que fue la que se usó para la creación del archivo con la tabla de particiones.

```
sfdisk /dev/sda < backup_sda.sfdisk
sfdisk /dev/sdb < backup_sdb.sfdisk
```

Los argumentos usados son:

/dev/sdX	Indica el dispositivo del cual se quiere clonar la tabla de particiones

11. Inicializar los volúmenes físicos que componen el grupo de volúmenes para ello se emplea la herramienta pvcreate.

```
pvcreate -u "Ds1dCd-Ow6m-nnyY-2jcI-MDtq-Rw1n-BZcpfB" --restorefile
volumenDatosConfig /dev/sda1
pvcreate -u "Ya1SlF-Rmif-8vQd-yfSd-twiM-77su-Z3VKai" --restorefile
volumenDatosConfig /dev/sdb1
```

Los argumentos usados son:

-u "UUID_particio_sdX1"	Indica el UUID que se le asignará a la partición cuando se inicialice el volumen físico.
--restorefile volumenDatosConfig	Indica el fichero de configuración obtenido con la herramienta vgcfgbackup. Esto se realiza para que los metadatos que se generan al recrear el volumen lógico sean iguales a los que existían cuando se clonó el grupo de volúmenes.
/dev/sdX1	Indica la partición en la que iniciará el volumen lógico.

12. Recrear el grupo de volúmenes volumenDatos para ello se emplea la herramienta vgcfgrestore.

```
vgcfgrestore -f volumenDatosConfig volumenDatos
```

Los argumentos usados son:

-f volumenDatosConfig	Indica el fichero de configuración obtenido con la herramienta vgcfgbackup.
volumenDatos	Indica el nombre del grupo de volúmenes que se van a recrear. Esto se debe añadir porque en el fichero de configuración puede haber varios grupos de volúmenes.

13. Activar todos los volúmenes lógicos de los grupos de volúmenes que existan en el sistema. Se utilizará la herramienta vgchange.

```
vgchange -a y
```

El argumento usado es:

-a y	Indica que el estado de los volúmenes lógicos cambia a disponible.

14. Restaurar el contenido de la partición con datos. En este ejemplo serán /dev/mapper/volumenDatos-root y /dev/sda1. Para ello se usará la herramienta fsarchiver de nuevo.

```
fsarchiver -j2 -v restfs root_backup.fsa
id=0,dest=/dev/mapper/volumenDatos-root
fsarchiver -j2 -v restfs boot_backup.fsa id=0,dest=/dev/sda2
```

Los argumentos usados son:

-j2	Indica el número de hilo que se utilizará en el proceso de clonación. Permite que el proceso se realice más rápido si se dispone de un procesador con varios núcleos. El número 2 nos informa de que se usarán 2 núcleos.
-v	Activa el modo verboso para poder ver el progreso de la clonación.
restfs	Indica que se va a realizar la restauración de un sistema de archivos.
X_backup.fsa	Indica el archivo que se usará como origen para la restauración.

id=0	Indica que se restaurará el primer sistema de archivo que se encuentre en este.
dest=/dev/X	Indica el destino sobre el que se realizará la restauración.

15. Recrear la partición swap en /dev/mapper/volumenDatos-swap. Para ello se usará la herramienta mkswap.

```
 mkswap -U "86379206-3568-48ab-bf55-d87e46fd337c" -L "servidor2-swap"
/dev/mapper/volumenDatos-swap
```

Los argumentos usados son:

-U "86379206-3568-48ab-bf55-d87e46fd337c"	Indica el UUID que se le asignará a la partición swap una vez que se cree.
-L "servidor2-swap"	Indica la etiqueta que se le asignará a la partición swap una vez se cree. Si no la tuviera antes, este argumento se podría omitir.
/dev/mapper/volumenDatos-swap	Indica el dispositivo en el que se creará la partición swap

16. Reinstalar el gestor de arranque grub. Este paso es un método alternativo al indicado en el paso 8 del proceso de restauración del RAID software. Se utilizan las herramientas mkdir, mount, y chroot[17], vgchange y grub2-install.

```
mkdir datos
mount /dev/mapper/centos-root datos/
mount /dev/sda1 datos/boot/
mount --bind /dev/ datos/dev
mount --bind /proc/ datos/proc
chroot datos/
vgchange -a y
grub2-install /dev/sda
grub2-install /dev/sda
```

El argumento usado para la herramienta mkdir es:

datos	El nombre del directorio que se quiere crear.

Los argumentos usados para mount cuando tiene dos argumentos son:

/dev/X	Indica el dispositivo que va a montar.
Y	Indica el lugar dónde realizar el montaje del dispositivo

Los argumentos usados para mount cuando posee tres argumentos son:

--bind	Indica que el subárbol de directorios montado en una ubicación se recree en otra para que esté disponible en ambos.
/dev/X	Indica la ruta desde donde comenzar el subárbol de directorios que será replicado.
Y	Indica el lugar dónde realizar el montaje del subárbol de directorios.

El argumento usado para la herramienta chroot[17] es:

datos/	Indica el directorio que se usará como directorio raíz.

El argumento usado para la herramienta vgchange es:

-a y	Indica que se debe cambiar el estado de los volúmenes lógicos a disponible.

El argumento usado para el comando grub2-install es:

/dev/sdX	Indica el dispositivo en cuyo sector de arranque se instalará el GRUB.

2.2.3 **Línea de continuación**

Una posible línea de continuación puede ser estudiar si el uso de versiones posteriores de LVM puede sustituir el uso de RAID sin perjudicar el rendimiento de escritura/lectura y manteniendo la redundancia. Actualmente las herramientas que gestionan LVM permiten crear volúmenes lógicos RAID. La desventaja de esto es que pierde rendimiento y deja de ser posible redimensionar estos volúmenes.

2.3 Copias de seguridad incrementales basado en git

En este apartado se detallará un procedimiento para realizar copias de seguridad incrementales basado en el control de versiones git[18] de manera periódica y programada.

La herramienta git es un control de versiones distribuido. Registra los cambios realizados sobre un conjunto de archivos a lo largo del tiempo y crea un histórico mediante el cual es posible recuperar versiones específicas más adelante. Es una herramienta muy útil ya que permite acceder a cambios que se hayan realizado sobre los archivos. Al tratarse de un sistema distribuido, se puede tener copias de los cambios realizados en varios sistemas; de esta manera, es posible recuperar la información, aunque un cliente o servidor haya sido eliminado.

En el

ANEXO C: Copias de seguridad incrementales basado en git se explica algunos conceptos útiles para entender la forma de trabajar de git y describe los estados por los que pasa un archivo hasta que se incluye en el histórico de cambios.

2.3.1 Situación actual y problemática

En el presente, en el laboratorio de Telemática se están realizando las copias de seguridad de aquellos ficheros esenciales para el correcto funcionamiento de este. Algunos ejemplos son las configuraciones de los servicios y aplicaciones usadas.

El procedimiento que se está usando actualmente consiste en la compresión mediante la herramienta tar. El objetivo es crear una copia de seguridad de los ficheros manteniendo sus propiedades. El siguiente paso sería comprimir estos archivos copiados con el fin de reducir su tamaño. En último lugar, se nombraría el elemento generado para identificar cuando se hizo dicha copia.

Este procedimiento tiene una serie de inconvenientes:

- Aumenta el espacio de almacenamiento consumido rápidamente. El motivo es que con cada copia realizada se duplicarán los ficheros, aunque no haya cambiado entre copias consecutivas.
- Es lento en la comparación de los cambios realizados en los ficheros entre diferentes copias. Esto se debe a que el proceso que se sigue es descomprimir, en primer lugar, ambas copias de seguridad y posteriormente, utilizar alguna herramienta como diff para la comparación.
- No se dispone de un control de los ficheros que cambiaron entre diferentes copias, por lo que debe comprobarse manualmente cual era el contenido de las copias y qué cambios sufrieron. Esto dificulta la búsqueda de una versión concreta en un conjunto de ficheros.
- No se realizan de forma automática, lo que conlleva a que su regularidad no sea constante. El procedimiento actual es lanzar un script manualmente pudiendo perderse cambios e impidiendo recuperar un estado donde el servicio o aplicación funcionara correctamente, ya que no se habría llevado a cabo en ese momento una copia de seguridad.
- Las copias realizadas se mantienen en local y es necesario transferirlas a mano a otros sistemas. Así se conseguiría que la redundancia sea mayor y se evitaría posibles pérdidas si en algún momento cae el sistema que lo almacena.

Debido a estos inconvenientes se valoran entre diferentes herramientas para poder diseñar un procedimiento que puede solventarlos a todos.

Una de las herramientas valoradas fue rsync[19]. Esta permite la realización de copias en remoto y la creación de copias incrementales. En estas últimas se copiarían aquellos ficheros cuyo contenido cambiaran. Para aquellos que no cambien, se crearían un enlace duro que reduciría el tamaño de las copias considerablemente. Sin embargo, esta opción fue desechada en pro del uso de git. Rsync [19] no permite llevar un control de los ficheros que cambian, ni de la manera en la que se modifican a lo largo de las diferentes copias. Además, el uso de enlace duro como método para reducir el espacio de almacenamiento consumido implicaba que si se modificaba un fichero perteneciente una copia antigua modificaría todas las copias asociadas, haciendo inservible la copia de seguridad.

2.3.2 Solución elegida

A continuación, se detallará el sistema diseñado para realizar copias incrementales para ello se hará uso del escenario donde será usado en el laboratorio.

El escenario estaría compuesto por dos sistemas donde el cliente sería, además, el anfitrión de la máquina virtual KVM. Esta última será el segundo sistema, que actuaría como servidor. El nombre para el repositorio de git será BackupGIT, tanto en el cliente como en el servidor.

Además, se asume la existencia de un usuario llamada usuarioBackup, propietario del directorio donde se aloja el repositorio de git, tanto en el servidor como en el cliente. El motivo de su uso es evitar que un usuario sin

privilegios edite el repositorio de git y lo borre o corrompa la información que almacena.

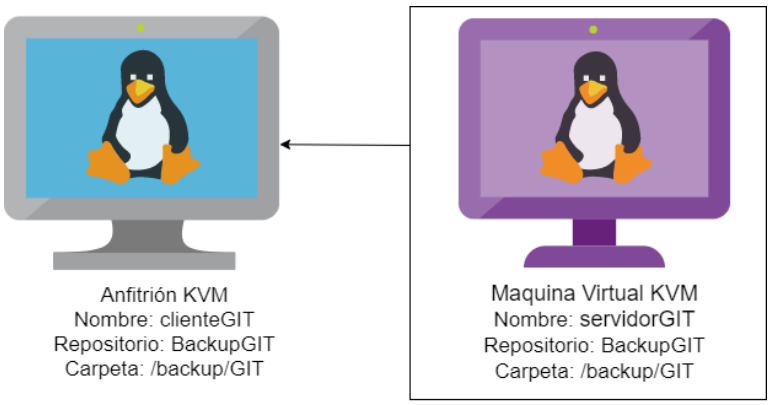

Ilustración 2-12 Escenario propuesto para las copias de seguridad incrementales con git

El funcionamiento del sistema se divide en cliente y servidor.

De las todas las tareas del servidor se explicarán la creación del repositorio de git, los ficheros para los cuales se quiere realizar la copia de seguridad, la confirmación de los cambios en los ficheros y la reducción del tamaño del repositorio y del histórico de cambios.

Se describirán las siguientes tareas del cliente: descarga y sincronización del repositorio git del servidor, descarga de los cambios existentes del repositorio, subida de cambios al repositorio de git.

La parte servidor, en la cual se detallará el proceso de crear un repositorio. realizar las copias, subidas y confirmación de cambios y por último el proceso por el cual se va a administrar el servidor para controlar su tamaño y eliminar información demasiado desactualizada o que ya no es útil.

Las herramientas usadas serán:

- git para trabajar con el sistema de control de versiones.
- rsync lleva a cabo las copias de los elementos deseados de forma automática y manteniendo sus propiedades en el repositorio.
- ssh[20] transfiere información con git entre el cliente y el servidor. Se necesitará también de ssh-keygen[21] y ssh-copy-id[22] para la creación de una clave pública para la conexión con ssh. Así, no será necesario la intervención humana a la hora de realizar la actualización o la subida de cambios desde el cliente.

2.3.2.1 Servidor

En el escenario propuesto, el servidor de git se alojará en la máquina virtual de nombre servidorGIT. Todas las acciones descritas en esta sección se desarrollarán sobre este. También se parte de haber iniciado sesión como superusuario para poder realizar la asignación de permisos necesarios.

En el

ANEXO C: Copias de seguridad incrementales basado en git se detallan los scripts y la forma de configurar la herramienta cron[23], todo ello para realizar los siguientes procedimientos, excepto la creación del repositorio de manera automatizada y programada.

2.3.2.1.1 Creación de repositorio

El siguiente procedimiento se podrá realizar mediante el script configurarRepositorio.sh que se detallará en el

ANEXO C: Copias de seguridad incrementales basado en git.

Los pasos que seguir son los siguientes:

1. Compruebar que exista el usuario usuarioBackup. A este se le asignará el directorio que contiene el repositorio de git con la herramienta id[24] .

```
id usuarioBackup
```

El argumento usado es:

ususarioBackup	Indica el usuario para el que se quiere comprobar su existencia.

2. Crear el directorio donde se alojará el repositorio de git. Se usará mkdir.

```
mkdir -p /backup/GIT
```

Los argumentos usados son:

-p	Indica que, si los directorios padres del directorio que se quiere crear no existieran, se crearían en el proceso sin provocar error.
/backup/GIT	Indica el directorio que se quiere crear. Para este caso se ha indicado la ruta absoluta.

3. Iniciar la estructura del fichero necesaria para el repositorio de git para ello se hará uso de la herramienta git.

```
git init /backup/GIT
```

Los argumentos usados son:

init	Indica que se cree un repositorio de git, con ello crea la estructura de ficheros necesario para ello.
/backup/GIT	Indica el directorio donde se quiere crear el repositorio de git. De usar este argumento se usará el directorio de trabajo como directorio donde crear el repositorio.

4. Acceder a la carpeta donde se encuentra alojado el repositorio /backup/GIT/, para ello se hará uso de la herramienta cd[25].

```
cd /backup/GIT/
```

Los argumentos usados son:

/backup/GIT/	Indica la ruta del directorio donde se quiere cambiar el directorio de trabajo

5. Configurar el repositorio para poder almacenar y mantener los metadatos asociado a los ficheros para

ello se hará uso de la herramienta git-store-meta[26].

```
git-store-meta.pl -i -f mtime,atime,mode,user,group,uid,gid,acl
```

Los argumentos usados son:

-i	Indica que se genere los procesos en el repositorio para antes de realizar confirmaciones de cambios se genere un fichero con los metadatos que se subirá al repositorio o tras actualizar los cambios se aplique los metadatos los ficheros y carpetas del repositorio.
-f mtime,atime,mode,user,group,uid,gid,acl	Indica qué metadatos se desea almacenar. En este caso se almacenará la fecha de la última modificación, la fecha del último acceso, los permisos de Unix, el nombre de usuario propietario, el nombre de grupo propietario, el ID de usuario propietario, el ID de grupo propietario y los ACLs.

6. Crear por primera vez el fichero .git_store_meta que guarda los metadatos de los ficheros o carpetas del repositorio. Es necesario realizarlo al menos la primera vez el resto de las ocasiones se realizará de manera automática. Para ello se usará de la herramienta git-store-meta.

```
git-store-meta.pl -s -f mtime,atime,mode,user,group,uid,gid,acl
```

Los argumentos usados son:

-s	Genera el fichero .git_store_meta que contendrá los metadatos de los ficheros y las carpetas del repositorio.
-f mtime,atime,mode,user,group,uid,gid,acl	Indica cuales son los metadatos que se desea almacenar. En este caso se guardará la fecha de última modificación, la fecha de último acceso, los permisos de Unix, el nombre de usuario propietario, el nombre de grupo propietario, el ID de usuario propietario, el ID de grupo propietario y el ACLs.

7. Realizar el primer commit para indicar cuando se creó el repositorio para ello se hará uso de la herramienta git.

```
git commit --allow-empty -a -m "Mensaje"
```

Los argumentos usados son:

commit	Confirma los cambios que existieran en el área de preparación. En este caso, al haberse creado un repositorio vacío, no se confirmará ningún cambio.
--allow-empty	Permite realizar una confirmación de cambios sin que exista realmente ningún cambio.
-m “Mensaje”	Añade el comentario con el que se quedará registrado la confirmación de cambios en el histórico.

8. Nombrar el repositorio para poder acceder a él desde el cliente usando git. Para este paso el directorio de trabajo deberá ser /backup/GIT/.

```
git remote add BackupGIT localhost:/backup/GIT/
```

Los argumentos usados son:

remote add	Añade una nueva referencia remota del repositorio.
BackupGIT	Indica el nombre que identificará la referencia remota del repositorio.
localhost:/backup/GIT/	Indica la ubicación de la referencia remota. En este caso, como lo que se quiere es que el repositorio local también sea reconocido por el nombre BackupGIT, se introducirá la dirección localhost junto a la ubicación de la carpeta que aloja el repositorio.

9. Cambiar el propietario del directorio donde se encuentra el repositorio y su contenido. El nuevo usuario será usuarioBackup. Se consigue con chown[27].

```
chown -R usuarioBackup /backup/GIT/
```

Los argumentos usados son:

-R	Indica que el cambio de propietario de fichero se hace recursivamente.
usuarioBackup	Es el usuario que será el propietario de los archivos.
/backup/GIT/	Ruta del directorio al que se cambiará de propietario. Si se hace de forma recursiva también cambiará de contenido.

2.3.2.1.2 Preparación de ficheros para realizar la copia de seguridad

El siguiente procedimiento se podrá realizar mediante el script copiarFicherosCarpetasServidor.sh que se detallará en el

ANEXO C: Copias de seguridad incrementales basado en git.

Los pasos que seguir son:

1. Realizar la copia de los ficheros o directorios que se quieren incluir en la copia de seguridad con la herramienta rsync.

```
rsync -acPRAX --delete-after rutaAbsoluta /backup/GIT
```

Los argumentos usados son:

-a	Opción para crear una copia recursivamente de los archivos que se encuentran dentro del directorio de rutaAbsoluta, preservando las propiedades, la fecha de modificación, los propietarios y los grupos de estos. Además, también se preservarán los enlaces simbólicos y se recreará el fichero de dispositivos y de especiales como sockets o fifos. Es equivalente a usar los argumentos -rlptgoD.
-c	Indica que para comprobar si un fichero cambio entre los ficheros a copiar y los que se encuentra ya en el directorio destino mediante el uso de una suma de comprobación. Si no se detectan cambios entre las sumas de comprobación, no se realizaría la copia de ese fichero o directorio.
-P	Opción para ver el progreso de una copia y reanudarla en caso de que se interrumpiera. Es equivalente a usar los argumentos --partial --progress.
-R	Sirve para que en el destino se replique el conjunto de ficheros y directorio tal y como se encuentran en el origen, conservando la ruta tal y como se le indicó. De esta manera si se introduce una ruta absoluta se replicará todo lo que contenga en el directorio de destino.
-A	Indica que se preserven los ACLs de los ficheros o directorios que se copien.
-X	Indica que se preserven los atributos extendidos de los ficheros o directorios que se copien.
--delete-after	Indica que se elimine como último paso en la copia aquellos archivos que no se encontraba en el origen cuando este es un directorio y se indicó que se copiara recursivamente.
rutaAbsoluta	Indica la ruta absoluta de un fichero o directorio origen que se copiará completamente en el destino indicado.
/backup/GIT	Indica el directorio destino donde se alojará la copia. Al tratarse de un destino local solo se indica la ruta absoluta.

Este paso se deberá llevar a cabo con cada fichero o directorio que se quiera incluir en la copia. En el caso del directorio es necesario que la ruta absoluta termine en "/" para que se copie de manera recursiva.

2. Eliminar los ficheros o directorios que ya no formen parte de la copia de seguridad. Se consigue con rm[28].

```
rm -Rf /backup/GIT/rutaAbsoluta
```

Los argumentos usados son:

-R	La eliminación se hará de manera recursiva si se indica como argumento un directorio que se vaya a borrar.
-f	Fuerza el eliminado, es decir, no se solicitará confirmación para realizar la eliminación y se ignora si el fichero o directorio no existe.
/backup/GIT/rutaAbsoluta	Indica el fichero o directorio que se quiere eliminar.

2.3.2.1.3 Realización de la confirmación de cambios

El siguiente procedimiento se podrá realizar mediante el script realizarCommitRepositorioServidor.sh que se detallará en el

ANEXO C: Copias de seguridad incrementales basado en git.

Los pasos que se siguen son los siguientes:

1. Acceder a la carpeta donde se encuentra alojado el repositorio /backup/GIT/ mediante cd.

```
cd /backup/GIT/
```

El argumento usado es:

/backup/GIT/	Indica la ruta del directorio donde se quiere cambiar el directorio de trabajo

2. Añadir todos los archivos que se encuentran en el directorio de trabajo al área de preparación. Después se confirman aquellos cambios que hayan sufrido los ficheros, ya sean cambios en su contenido o en las propiedades, o si se añade alguno nuevo o se elimina. Para ello se hará uso de la herramienta git.

```
git add -A
```

Los argumentos usados son:

add	Añade los ficheros del directorio al área de preparación para la siguiente confirmación de cambios.
-A	Indica que se añada todos los ficheros que contenga el directorio de trabajo.

3. Compruebar que exista cambios en el área de preparación, ya que si no existiera ningún cambio el siguiente paso fallaría. Para ello se hará uso de la herramienta git.

```
git diff --cached
```

Los argumentos usados son:

diff	Comprueba si existen cambios entre la zona de preparación y los archivos que se encuentran en el directorio de trabajo.
--cached	Indica que se cambie los archivos a comparar, se comparará la zona de preparación y los archivos que se encontraba en la última confirmación de cambios del repositorio local.

Si la ejecución del comando anterior arrojara algún resultado entonces se podría continuar en el procedimiento. En caso contrario, el procedimiento acabaría aquí, ya que no habría ningún cambio a confirmar.

4. Realizar la confirmación de los cambios para ello se hará uso de la herramienta git.

```
git commit -a -m "Backup FECHA"
```

Los argumentos usados son:

commit	Confirma los cambios que existen en el área de preparación. En este caso, al haberse creado un repositorio vacío, no se confirmará ningún cambio.
-a	Indica que se incluyan todos los cambios que se encuentren en la zona de preparación a la confirmación de cambios que se va a realizar.
-m "Backup FECHA"	Añade el comentario con el que se quedará registrado la confirmación de cambios en el histórico.

2.3.2.1.4 Reducción del histórico y optimización de repositorio

En este procedimiento lo que se realiza es la eliminación de todas las confirmaciones de cambios que se hicieron anterior a una fecha determinada. Tras ello, se llevarán a cabo varias tareas de mantenimiento del repositorio.

El siguiente procedimiento se podrá realizar mediante el script borrarCommitAnterioresAFecha.sh que se detallará en el

ANEXO C: Copias de seguridad incrementales basado en git.

Los pasos que seguir son los siguientes:

1. Acceder a la carpeta donde se encuentra alojado el repositorio /backup/GIT/ mediante cd.

```
cd /backup/GIT/
```

El argumento usado es:

/backup/GIT/	Indica la ruta del directorio donde se quiere cambiar el directorio de trabajo

2. Buscar la primera confirmación de cambios posterior a la fecha que se quiera como límite para mantener en el histórico de cambios para ello se hará uso de la herramienta git.

```
git rev-list -1 --before="tiempoARetroceder" --date=relative master
```

Los argumentos usados son:

rev-list	Muestra los identificadores únicos de las confirmaciones de cambios realizadas en orden cronológico.
-1	Indica que solo se muestre un resultado.
--before="tiempoARetroceder"	Indica el periodo de tiempo que se quiere retroceder para tomar como punto inicial para la lista. El formato es 'X Y ago' siendo Y month, year, day, hour, minute o second y X el número de divisiones de tiempo que se quiere retroceder.
--date=relative	Indica que el formato del argumento before es relativo a la fecha actual del sistema, es decir, se aplicará lo que señale la fecha actual para obtener la fecha límite. Sería como ejecutar la herramienta date[29]: date -d "tiempoARetroceder"
master	Rama pertenecerán los identificadores mostrados.

Este comando proporcionará un identificador. A partir de este punto, se identificará por HASH.

3. Crear una rama temporal con el nombre rama_temporal cuyo comienzo será la confirmación de cambios identificado con HASH. Se usará la herramienta git.

```
git checkout --orphan rama_temporal HASH
```

Los argumentos usados son:

checkout	Cambia el espacio de trabajo para que coincida con la versión almacenada en la rama.
--orphan	Crea una nueva rama desde una determinada confirmación de cambios.

rama_temporal	Nombre de la nueva rama que se creará.
HASH	Indica la confirmación de cambios que se tomará como punto inicial de la nueva rama.

Tras la ejecución del comando el repositorio quedará de la siguiente manera:

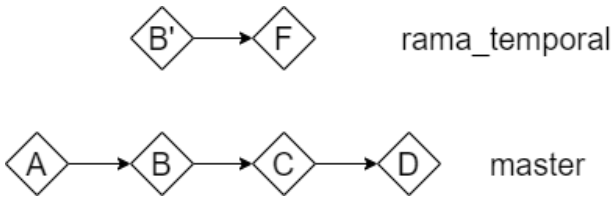

Ilustración 2-13 Diagrama del estado del repositorio tras crear la rama temporal

Siendo B y B' diferentes confirmaciones de cambios con el mismo contenido.

4. Realizar una confirmación de cambios sobre la rama rama_temporal, cuya finalidad es confirmar la versión tal y como se encontraba en la confirmación de cambios HASH. Así se puede identificar en el historial el momento en el que se hizo el borrado de las confirmaciones antiguas. Para ello, se utiliza la herramienta git.

```
GIT_COMMITTER_DATE="FECHA_PARA_COMMIT" git commit --date
"FECHA_PARA_COMMIT" -m "Truncado del historial antes de la
FECHA_PARA_COMMIT"
```

Los argumentos usados son:

GIT_COMMITTER_DATE="FECHA_PARA_COMMIT"	Fecha que se usará para calcular la marca de tiempo en el campo de la confirmación de cambio. Gracias a esto se podrá ordenar cronológicamente y hacer que aparezca esta confirmación de cambio como la más antigua, aunque sea esta la última confirmación de cambios realizada.
commit	Confirma los cambios que se situaban en el área de preparación. En este caso, al haberse creado un repositorio vacío, no se confirmará ningún cambio.
--date "FECHA_PARA_COMMIT"	Fecha que se usará para calcular la marca de tiempo en el campo de la confirmación de cambio. Gracias a esto se podrá ordenar cronológicamente y hacer que aparezca esta confirmación de cambio como la más antigua, aunque sea esta la última confirmación de cambios realizada.
-m "Truncado del historial antes de la FECHA_PARA_COMMIT"	Confirma los cambios que se situaban en el área de preparación. En este caso, al haberse creado un repositorio vacío, no se confirmará ningún cambio.

La variable FECHA_PARA_COMMIT se obtendrá mediante el uso de la herramienta date para ser coherente con el paso 2:

```
date -d "tiempoARetroceder"
```

5. Fuerzar que el comienzo de la rama máster(principal) coincida con el inicio de la rama rama_temporal. De esta manera se elimina la referencia de las confirmaciones de cambios antiguos del histórico. Para ello se hará uso de la herramienta git.

```
git rebase --committer-date-is-author-date --onto rama_temporal
HASH master
```

Los argumentos usados son:

rebase	Aplica los cambios realizados sobre una determinada rama en otra rama.
--committer-date-is-author-date	Hace uso de la fecha que tuviera la confirmación de cambios para aplicar en la operación.
--onto	Indica que se modifique el histórico para que la confirmación de cambios inicial de una rama cambie.
rama_temporal	Indica la rama origen desde donde se indicará que comienza la rama destino.
HASH	Indica la confirmación de cambios que pasará a ser el comienzo de la rama destino. Se sustituirá por las confirmaciones que hubiera en la rama origen.
master	Indica la rama destino de la operación.

Tras la ejecución del comando, el repositorio quedará de la siguiente forma:

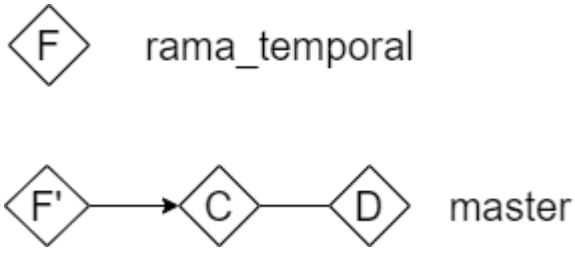

Ilustración 2-14 Diagrama del estado del repositorio tras forzar el nuevo comienzo del repositorio

Siendo F y F' diferentes confirmaciones de cambios con el mismo contenido.

6. Eliminar la rama rama_temporal para poder repetir el procedimiento varias veces sin cambiar el nombre, además, la rama dejará de ser útil. Se lleva a cabo con git.

```
git branch -D rama_temporal
```

Los argumentos usados son:

branch	Indica que se va a realizar una operación sobre una rama
-D	Indica que se va a eliminar la rama seleccionada
rama_temporal	Indica el nombre de la rama a eliminar

Tras la ejecución del comando, el repositorio quedará de la siguiente manera:

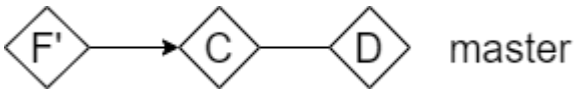

Ilustración 2-15 Diagrama del estado del repositorio tras borrar la rama temporal

7. Eliminar todos los objetos sin referencia de la base de datos del repositorio usando git.

```
git prune --progress
```

Los argumentos usados son:

branch	Indica que se elimine los objetos sin referencia de la base de datos.
--progress	Muestra el progreso del proceso.

8. Eliminar los ficheros innecesarios y se optimiza el repositorio con git.

```
git gc --aggressive
```

Los argumentos usados son:

gc	Inicia el recolector de basura y optimiza el repositorio.
--aggressive	Realiza el proceso de forma más intensiva. Esto provoca que se requiera más tiempo para finalizarlo.

2.3.2.2 Cliente

El cliente de git se alojará en el host de nombre clienteGIT del escenario propuesto. Durante esta sección todas las acciones se realizarán sobre este. También se parte de haber iniciado sesión como superusuario para poder realizar la asignación de permisos necesarios.

Para algunos de los procedimientos que se explicarán, será necesario acceder al servidor de git. Para conseguirlo, se hará uso de la herramienta ssh junto con la variable de entorno GIT_SSH. Esto permitirá configurar la conexión para poder adaptarlo a nuestro escenario. En esta variable, se debe indicar la localización del ejecutable que será usado para establecer la conexión mediante ssh.

El script que se usará será el siguiente y se identificará por ejecutableSSH:

```
#!/bin/sh
ssh -i /var/clavePublicaRoot -oPort=2222 -obatchmode=yes $*
```

Los argumentos usados para la herramienta ssh:

-i /var/clavePublicaRoot	Se usará una clave pública para establecer la conexión. Es necesario configurarla, por lo que se proporciona un script, configurarSshConexionConClavePublica.sh, que se encuentra en el ANEXO C: Copias de seguridad incrementales basado en git.
-oPort=2222	Define el puerto que tiene expuesto el servidor para la conexión mediante ssh. Es necesario si el puerto ssh es distinto a 22.
-obatchmode=yes	Indica que la autentificación de la conexión no se hará de forma iterativa. La ejecución fallará si es necesario introducir una contraseña o la passphrase de la clave pública.
$*	Se suministrarán como argumento todos los argumentos introducidos al script como una cadena de texto. Esto es necesario porque la herramienta git proporcionará los argumentos que sean necesarios en cada momento.

Los procedimientos del cliente que requieren una conexión mediante ssh están ideados para usarlos con una clave pública. Para configurar la clave pública de la conexión ssh se proporciona el script configurarSshConexionConClavePublica.sh que se detallará en el

ANEXO C: Copias de seguridad incrementales basado en git.

En el

ANEXO C: Copias de seguridad incrementales basado en git se detallarán los scripts y cómo configurar la herramienta cron para realizar los próximos procedimientos, excepto la clonación del repositorio de manera automatizada y programada.

2.3.2.2.1 Clonación del repositorio de git

El siguiente procedimiento se podrá realizar mediante el script configurarCliente.sh que se detallará en el

ANEXO C: Copias de seguridad incrementales basado en git.

Los pasos que siguen son los siguientes:

1. Compruebar que el usuario usuarioBackup al que se asignará al directorio que contiene el repositorio de git para ello se usara la herramienta id.

```
id usuarioBackup
```

El argumento usado es:

ususarioBackup	Indica el usuario para el cual se quiere comprobar su existencia.

2. Crear el directorio donde se alojará el repositorio de git con la herramienta mkdir.

```
mkdir -p /backup/GIT
```

Los argumentos usados son:

-p	Indica que, si los directorios padres del directorio que se quiere crear no existieran, se crearían en el proceso sin provocar error.
/backup/GIT	Indica el directorio que se quiere crear. Para este caso se ha escrito la ruta absoluta.

3. Acceder a la carpeta donde se encuentra alojado el repositorio /backup/GIT/ mediante cd.

```
cd /backup/GIT/
```

Los argumentos usados son:

/backup/GIT/	Indica la ruta del directorio donde se quiere cambiar el directorio de trabajo

4. Clonar el repositorio de git desde el servidorGIT, para ello se hará uso de la herramienta git.

```
GIT_SSH=ejecutableSSH git clone usuarioBackup@servidorGIT:
/backup/GIT -o BackupGIT .
```

Los argumentos usados son:

clone	Crea un repositorio de git. Con ello genera la estructura de ficheros necesarios.
usuarioBackup@servidorGIT: /backup/GIT	Indica el usuario remoto propietario del directorio donde se encuentra el repositorio, la dirección remota del servidor y la ubicación remota donde se encuentra el repositorio. Este argumento indica contra quien se debe establecer la conexión ssh.
-o BackupGIT	Indica el nombre del repositorio remoto que clonará y con el que se hará

	referencia.
.	Indica que el repositorio que se clonará es el directorio de trabajo actual.

5. Acceder a la carpeta donde se encuentra alojado el repositorio /backup/GIT/ mediante cd.

```
cd /backup/GIT/
```

Los argumentos usados son:

/backup/GIT/	Indica la ruta del directorio donde se quiere cambiar el directorio de trabajo

6. Configurar el repositorio para poder almacenar y mantener los metadatos asociados a los ficheros usando git-store-meta.

```
git-store-meta.pl -i -f mtime,atime,mode,user,group,uid,gid,acl
```

Los argumentos usados son:

-i	Genera los procesos para el repositorio. Antes de realizar las confirmaciones de cambios se genera un fichero con los metadatos que se subirán al repositorio. o tras actualizar los cambios se aplique los metadatos los ficheros y carpetas del repositorio.
-f mtime,atime,mode,user,group,uid,gid,acl	Indica cuáles son los metadatos que se desea almacenar. En este caso, se almacenará la fecha de última modificación, la fecha de último acceso, los permisos de Unix, el nombre de usuario propietario, el nombre de grupo propietario, el ID de usuario propietario, el ID de grupo propietario y el ACLs.

7. Cambiar el propietario al directorio donde se encuentra el repositorio y su contenido para que el propietario sea el usuario usuarioBackup. Para ello se hará uso de la herramienta chown.

```
chown -R usuarioBackup /backup/GIT/
```

Los argumentos usados son:

-R	Indica que el cambio de propietario de fichero se hace recursivamente.
usuarioBackup	Usuario que será el propietario de los archivos.
/backup/GIT/	Ruta del directorio al cual se le cambiará el propietario. Al hacerse recursivamente, también se cambia al contenido de este.

2.3.2.2.2 Actualización del repositorio

El siguiente procedimiento se podrá realizar mediante el script actualizarCliente.sh que se detallará en el

ANEXO C: Copias de seguridad incrementales basado en git.

Los pasos que seguir son los siguientes:

1. Acceder a la carpeta donde se encuentra alojado el repositorio /backup/GIT/ mediante cd.

```
cd /backup/GIT/
```

El argumento usado es:

/backup/GIT/	Indica la ruta del directorio donde se quiere cambiar el directorio de trabajo

2. eliminar las modificaciones que se encuentren en la carpeta local o en la zona de preparación del cliente. Estos cambios podrían implicar un conflicto a la hora de actualizar el histórico de cambios. Para ello se hará uso de la herramienta git.

```
GIT_SSH=ejecutableSSH git reset --hard BackupGIT
```

Los argumentos usados son:

reset	Eliminan los posibles cambios que hubiera en la zona de preparación.
--hard	Además de borrar los cambios en la zona de preparación, también se restablecerá el espacio de trabajo para que coincida con la última confirmación de cambios que se realizó.
BackupGIT	Indica sobre qué commit debe hacer la restauración de los ficheros. En este caso, al indicar el nombre del repositorio, estaremos indicado la última confirmación realizada sobre el repositorio remoto.

3. Descargar todas las confirmaciones de cambios realizada en el servidor y se actualiza el espacio de trabajo con la última confirmación de cambios que se realizó. Para ello se hará uso de la herramienta git.

```
GIT_SSH=ejecutableSSH git pull --no-edit BackupGIT
```

Los argumentos usados son:

pull	Descarga los cambios realizados en el repositorio remoto y los integra en el repositorio local.
--no-edit	Indica que no se abra el editor de texto para introducir el mensaje en caso de que fuera necesario realizar un mergeo de la información del repositorio local y el repositorio remoto. Esto generaría un mensaje automáticamente pero no es necesario ya que nunca habrá discrepancia entre los servidores que fueran necesaria de mergear debido al paso anterior
BackupGIT	Indica el nombre del repositorio remoto del que se obtendrá los cambios a descargar

	e integrar.

2.3.2.2.3 Preparación de ficheros para realizar la copia de seguridad

El siguiente procedimiento se podrá realizar mediante el script copiarFicherosCarpetasCliente.sh que se detallará en el

ANEXO C: Copias de seguridad incrementales basado en git. Los pasos que seguir serán los mismos que se detallaron en el apartado Preparación de ficheros para realizar la copia de seguridad.

2.3.2.2.4 Realización de la confirmación de cambio

El siguiente procedimiento se podrá realizar mediante el script realizarCommitRepositorioCliente.sh que se detallará en el

ANEXO C: Copias de seguridad incrementales basado en git. Los pasos que seguir serán los mismos que se detallaron en el apartado Realización de la confirmación de cambios. Lo único que cambia que será necesario realizar un último paso.

El último paso por realizar será actualizar los cambios realizados en el repositorio local en el repositorio remoto para ello se hará uso de la herramienta git.

```
GIT_SSH=ejecutableSSH git push BackupGIT master
```

Los argumentos usados son:

push	Indica que se haga la subida de los cambios del repositorio local al repositorio remoto.
BackupGIT	Indica el nombre del repositorio remoto donde se subirá los cambios.
master	Indica la rama sobre la que se aplicarán los cambios que hubiera en el repositorio local respecto al repositorio remoto.

2.3.2.3 Recuperación de una confirmación de cambios concreta

En esta sección se detallarán varios procedimientos para poder recuperar una determinada confirmación de cambios del repositorio o de un único fichero. Para ambos, será necesario conocer el identificador o hash de la confirmación de cambios. Se consigue con git.

```
git log
```

El argumento usado es:

log	Indica que se muestre el histórico de cambios

La ejecución de este comando es iterativa, permite buscar o recorrer las confirmaciones de cambios con los mismos controles que podrían usarse con la herramienta less[30].

El resultado del comando ejecutado será algo como la siguiente captura:

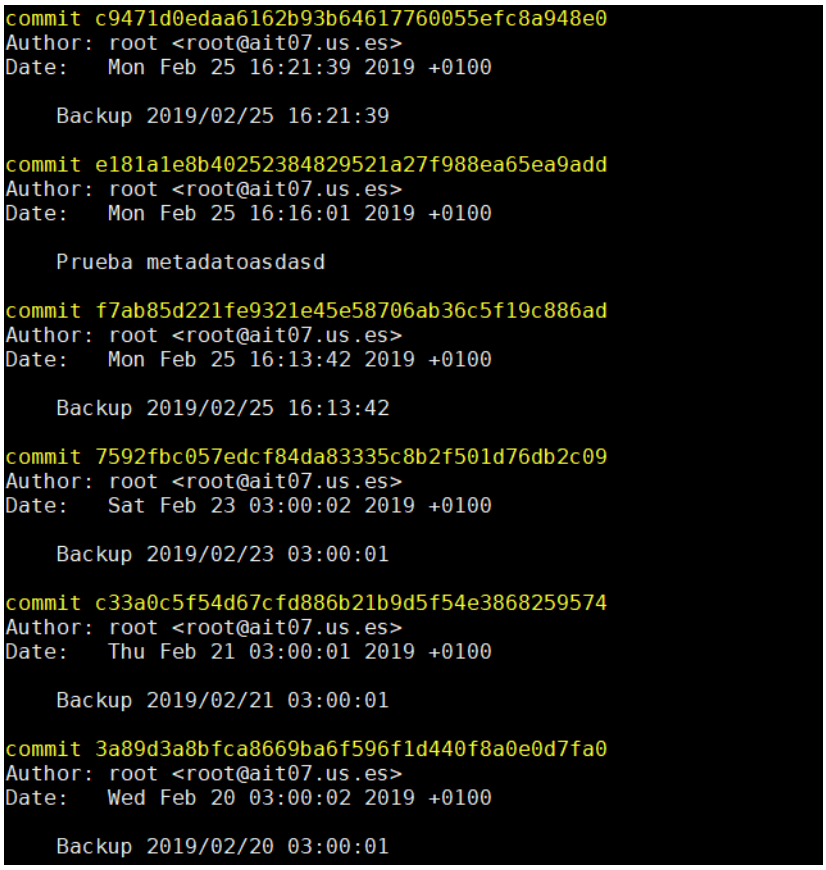

Ilustración 2-16 Resultado del comando git log

.

El identificador o HASH de la confirmación de cambios será el conjunto de letras y numero que aparecen a continuación de commit y debajo aparecerá la información relativa a dicha confirmación de cambios.

Un ejemplo sería el identificador de la confirmación de cambios 7592fbc057edcf84da83335c8b2f501d76db2c09 y la información asociada:

Author: root <root@ait07.us.es>

Date: Sat Feb 23 03:00:02 2019 +0100

Backup 2019/02/23 03:00:01

Una vez que se ha elegido la confirmación de cambios concreta para la recuperación una determinada versión, se puede optar por dos opciones:

- Recuperación del contenido de un único fichero usando la herramienta git.

```
git show HASHRevision:rutaFichero
```

La ejecución de este comando es iterativa, permite buscar o recorrer el fichero en la versión elegida con los mismos controles que se usan con la herramienta less. También es posible redirigir su salida estándar a un fichero concreto, volcando de esta manera el contenido del fichero en esa determinada versión.

Los argumentos usados son:

show	Muestra el contenido de un determinado fichero en una revisión o versión concreta.
HASHRevision	Indica la versión o confirmación de cambios sobre la que se quiere extraer el fichero especificado.
rutaFichero	Ruta absoluta del fichero que se quiere recuperar.

- Recuperación de una versión concreta del repositorio. Para ello se necesita de git.

```
git checkout HASHRevision
```

Los argumentos usados son:

checkout	Restaura el contenido del espacio de trabajo a una determinada versión o al contenido de una rama.
HASHRevision	Indica la versión o confirmación de cambios sobre la que se quiere extraer el fichero especificado.

Tras finalizar las operaciones necesarias de recuperación, se deberá volver a la última confirmación de cambios. Para ello, se hará uso de la herramienta git.

```
git checkout -f master
```

Los argumentos usados son:

checkout	Restaura el contenido del espacio de trabajo a una determinada versión o al contenido de una rama.
-f	Descarta los posibles cambios que hubiera en el área de preparación o en el espacio de trabajo. Se actualiza con el contenido de la rama.
master	Actualiza y hace referencia a la rama principal(master). Esto hace que se actualice con el contenido de la última confirmación de cambios.

2.3.2.4 Ver cambios realizado entre confirmaciones

En este esta sección se detallará el procedimiento para poder conocer los cambios que sufrió un determinado fichero entre dos confirmaciones de cambios.

Para comenzar, será necesario obtener el identificador o hash de las confirmaciones de cambios que se desean comparar y para ello, se usará la herramienta git junto con el argumento log, tal y como se detalla en la sección.

Se puede optar por dos opciones para conocer los cambios que se produjeron:

- Conocer los cambios que hubiera entre dos confirmaciones concretas. Se hará uso la herramienta git.

```
git diff HASH_1 HASH_2 rutaFichero
```

Los argumentos usados son:

diff	Restaura el contenido del espacio de trabajo a una determinada versión o al contenido de una rama.
HASH_1	Indica la confirmación de cambios que se usará con origen a la hora de realizar la comparación.
HASH_2	Indica la confirmación de cambios que se usará con destino a la hora de realizar la comparación. Si no se especifica, se usará la última realizada.
rutaFichero	Indica el fichero que se quiere comparar entre las dos confirmaciones de cambios. Si no se especifica, se compararían todos los ficheros del repositorio. También se puede especificar un directorio.

- Conocer los cambios que hubo en una determinada confirmación de cambios respecto a la anterior, para lo que se usará la herramienta git.

```
git show HASH_1
```

Los argumentos usados son:

show	Indica que se muestre los cambios que se produjeron en una determinada confirmación de cambios.
HASH_1	Indica la confirmación de cambios de la cual se quiere conocer los cambios realizados en ello.

2.3.3 **Líneas de continuación**

Las posibles líneas de continuación podrían ser las siguientes:

- Crear un script que permita la configuración de la herramienta cron de manera automatizada. De esta forma, se simplificaría la configuración del servidor y el cliente.

Permitir que tanto cliente como servidor puedan subir su contenido a ramas diferentes. Así, se tendría copias de seguridad tanto del cliente como del servidor en un mismo repositorio.

2.4 Sincronización de copias de seguridad mediante rsync

En esta sección se detallará un procedimiento por el cual se podrá disponer de copias de seguridad replicadas en dos o varios sistemas. Su fin es poder disponer de las copias de seguridad, aunque no se pueda acceder ni en remoto ni físicamente al sistema.

Para realizar este procedimiento se hará uso de la herramienta rsync. Esta herramienta permite hacer copias de ficheros, ya sea en remoto como en local, manteniendo ciertas propiedades como el usuario y grupo propietarios, permisos, sus ACLs o la fecha de última modificación o acceso.

2.4.1 Situación actual y problemática

Actualmente en el laboratorio de Telemática se están realizando se encuentra almacenada las copias de seguridad en un servidor únicamente y también se realiza copias de formar manuales parte de las copias de seguridad en almacenamiento externos para poder recuperar los servidores en caso de que esto quedarán corruptos.

Este procedimiento tiene una serie de inconvenientes:

- Se deben realizar manualmente las copias de seguridad para tener redundancia.
- No existe un control de las modificaciones de las copias de seguridad.
- No se mantienen los metadatos asociados a los ficheros y directorios, como pueden ser los permisos o los propietarios.

Debido a estos inconvenientes, se valora el uso de diferentes herramientas que pudieran solventarlos.

Entre las herramientas valoradas se encontraba scp[31] que permite automatizar las copias en sistemas remotos mediante el uso de conexiones ssh. Sin embargo, fue desechada en pro del uso de la herramienta rsync. Esta última seguía haciendo uso de conexiones ssh y, además, conservaba los metadatos asociados a los ficheros y directorios.

2.4.2 Solución elegida

A continuación, se detallará el sistema diseñado para realizar la sincronización de copias de seguridad para ello se hará uso del escenario donde será usado en el laboratorio.

El escenario está compuesto por dos sistemas donde el cliente será el servidor secundario y el servidor será el servidor principal. En el servidor principal se crearán y almacenarán las copias de seguridad. En el servidor secundario se almacenará un duplicado de las copias de seguridad.

Las diferentes copias de seguridad, los archivos y los directorios se encontrarán en diferentes carpetas que compartirán prefijo. El prefijo elegido para el escenario será copiaSeguridad_.

Además, se asume la existencia de un usuario, llamada usuarioBackup, que será el propietario del directorio donde se alojarán las copias de seguridad en el servidor y en el cliente. Se usa como remedio para evitar que un usuario sin privilegios pueda editar, borrar o corromper la información de las copias de seguridad.

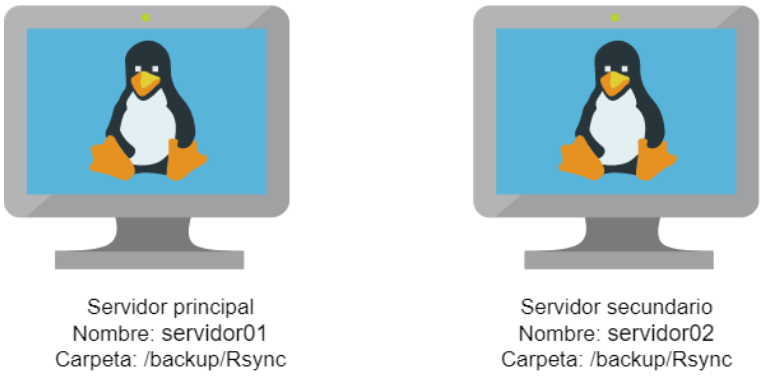

Ilustración 2-17 Escenario propuesto para la sincronización de copias seguridad

El funcionamiento del sistema se divide en cliente y servidor.

El servidor se encargará de realizar las siguientes acciones:

- Cambiar el propietario
- Convertir las copias de seguridad en inmutables, para impedir la modificación de estas incluso con permisos de superusuario si no se desactivan.
- Crear una suma de comprobación de las copias de seguridad para verificar posteriormente si se realizó alguna modificación sobre ellas.
- Comprobar si se modificó alguna copia de seguridad de la que se disponía anteriormente una suma de comprobación.

El cliente se encargará de realizar las siguientes acciones:

- Comprobar las sumas de comprobación contra el contenido que existe en el servidor.
- Comprobar que no cambia la copia del cliente, aunque fuese modificada en el servidor.
- Copiar las copias de seguridad que no estuvieran en el cliente.
- Crear una suma de comprobación de las copias de seguridad para verificar posteriormente si se realizó alguna modificación sobre estas.

Las herramientas usadas serán:

- rsync hará las copias de los elementos deseados de forma automática y manteniendo los metadatos asociados a las copias de seguridad.
- chattr[32] convertirá en inmutable los archivos y ficheros que componen las copias de seguridad.
- tar[20], [33] junto md5sum[34], serán las herramientas usadas para calcular las sumas de comprobación.
- ssh realizará la transferencia de información entre cliente y servidor.

2.4.2.1 Servidor

El sistema que actuará como servidor será el sistema nombrado como servidor01 del escenario propuesto. Durante esta sección todas las acciones se realizarán sobre este. También se parte de haber iniciado sesión como superusuario para poder completar algunas tareas como convertir en inmutables los ficheros o directorios que componen las copias de seguridad.

El procedimiento que se va a detallar a continuación se podrá realizar mediante el script scriptBackupCrearChecksumYConvertirInmutable.sh que se detallará en el

ANEXO D: Sincronización de copias de seguridad mediante rsync.

Los pasos que seguir son los siguientes:

1. Compruebar la existencia del usuario usuarioBackup, que será al que se le asignará el directorio que contiene el repositorio de git. Para ello se usará la herramienta id.

```
id usuarioBackup
```

El argumento usado es:

ususarioBackup	Hará la comprobación de que dicho usuario existe.

2. Acceder al directorio donde se encuentra alojada la copia de seguridad /backup/Rsync/. Para ello se hará uso de la herramienta cd.

```
cd /backup/Rsync/
```

Los argumentos usados son:

/backup/Rsync/	Indica la ruta del directorio donde se quiere cambiar el directorio de trabajo

3. Cambiar el propietario de los directorios donde se encuentran las copias de seguridad y su contenido. El nuevo usuario será usuarioBackup. Para ello se hará uso de la herramienta chown.

```
chown -R usuarioBackup rutaDirectorio
```

Los argumentos usados son:

-R	Indica que el cambio de propietario de fichero se haga recursivamente.
usuarioBackup	Indica el nuevo propietario de los archivos.
rutaDirectorio	Indica la ruta del directorio al cual se le cambiará el propietario. Al hacerse recursivamente, también se cambiará al contenido de este.

Este paso se deberá realizar con los directorios que comienzan por el prefijo "copiaSeguridad_".

4. Conviertir en inmutables los archivos y directorios que componen las copias de seguridad. Para ello se hará uso de la herramienta chattr.

```
chattr -R +i rutaDirectorio
```

Los argumentos usados son:

-R	Lleva a cabo los cambios de forma recursiva si se indica en un directorio.
+i	Convierte en inmutable el fichero o directorio indicado.
rutaDirectorio	Indica la ruta del directorio que contiene la copia de seguridad.

Este paso se deberá realizar con directorios que comienzan por el prefijo "copiaSeguridad_".

5. Calcular la suma de comprobación de los directorios que contienen las copias de seguridad. Para ello se hará uso de las herramientas tar y md5sum.

```
tar -cf - rutaDirectorio | md5sum > nombreDirectorioChecksum
```

Los argumentos usados son:

-c	Crea un nuevo fichero comprimido.
-f -	Vuelca el contenido sobre el fichero del nuevo archivo comprimido sobre la salida estándar en vez de sobre un fichero concreto.
rutaDirectorio	Indica la ruta del directorio al cual se le va a calcular la suma de comprobación.
nombreDirectorioChecksum	Indica el fichero en el cual se almacenará la suma de comprobación calculada.

Un ejemplo del cálculo de la suma de comprobación se puede ver en la siguiente captura.

```
root@lt-servidor2:/backup# tar -c /backup/Rsync/copiaSeguridad_host/ | md5sum > copiaSeguridad_hostChecksum
tar: Eliminando la `/' inicial de los nombres
root@lt-servidor2:/backup# cat copiaSeguridad_hostChecksum
251f083ffd897f4d5fe55e88316205ca  -
```

Este paso se deberá realizar con los directorios que comiencen por el prefijo "copiaSeguridad_".

6. Se comprueba que no hayan sido modificadas las copias de seguridad. Para ello, se parte de que existe la suma de comprobación anterior para los directorios que guardan las copias. Se utilizará la herramienta diff[35].

```
diff -q nombreDirectorioChecksumNuevo
nombreDirectorioChecksumAntiguo
```

Los argumentos usados son:

-q	Muestra un mensaje informando únicamente si los ficheros difieren en su contenido.
nombreDirectorioChecksumNuevo	Indica el fichero que contiene la suma de comprobación calculada.
nombreDirectorioChecksumAntiguo	Indica el fichero que contiene la suma de comprobación. Nos servirá como punto de referencia para saber si la copia de seguridad fue modificada.

Este paso se deberá realizar con directorio que comienza con el prefijo "copiaSeguridad_".

2.4.2.2 Cliente

El sistema que actuará como cliente será el sistema nombrado como servidor2 en el escenario propuesto. Durante esta sección todas las acciones se realizarán sobre este. También se parte de haber iniciado sesión como superusuario para poder completar ciertas tareas, como convertir en inmutables los ficheros y los directorios que componen las copias de seguridad.

El procedimiento que se van a detallar a continuación se podrá realizar mediante el script scriptBackupRemoto.sh que se detallará en el

ANEXO D: Sincronización de copias de seguridad mediante rsync.

Los pasos que seguir son los siguientes:

1. Compruebar la existencia del usuario usuarioBackup, que será al que se le asignará el directorio que contiene el repositorio de git. Para ello se usará la herramienta id.

```
id usuarioBackup
```

El argumento usado es:

ususarioBackup	Hará la comprobación de que dicho usuario existe.

2. Crear el directorio donde se alojará el repositorio de git. Para ello se usará la herramienta mkdir.

```
mkdir -p /backup/ Rsync
```

Los argumentos usados son:

-p	Con este argumento, si los directorios padres del directorio que se quiere crear no existieran, se crearían en el proceso sin provocar ningún error.
/backup/ Rsync	Indica el directorio que se quiere crear. Para este caso se ha indicado la ruta absoluta.

3. Acceder al directorio donde se encuentra alojada la copia de seguridad /backup/Rsync/. Para ello se hará uso de la herramienta cd.

```
cd /backup/Rsync/
```

El argumento usado es:

/backup/Rsync/	Indica la ruta del directorio donde se quiere cambiar el directorio de trabajo

4. Obtener de los directorios que contienen el prefijo "copiaSeguridad_" que se encuentra en el servidor para posteriormente conocer que directorios será necesario copiar al cliente. Para ello se hará uso de las herramientas ssh, cd, ls[36] y grep[37].

```
ssh -obatchmode=yes -oPort=2222 -i rutaClavePublica
usuarioBackup@servidor01 "cd /backup/Rsync/; ls -d * | grep
^copiaSeguridad_"
```

Los argumentos usados se dividirán en los argumentos usados con las diferentes herramientas.

Los argumentos usados con la herramienta ssh son:

-obatchmode=yes	La autentificación de la conexión no se hará de forma iterativa. La ejecución fallará si es necesario introducir contraseña o la passphrase de la clave pública.
-oPort=2222	Indica el puerto donde se expone el servidor para la conexión. Es necesario si el puerto ssh es distinto a 22.

-i rutaClavePublica	Se usará una clave pública para establecer la conexión.
usuarioBackup@servidor01	Indica el usuario y la dirección remotos que se usará para establecer la conexión ssh.
comando	Indica el script que se ejecutará en servidor01 cuando se establezca la conexión ssh.

El argumento usado con la herramienta cd es:

/backup/Rsync/	Indica la ruta del directorio donde se quiere cambiar el directorio de trabajo

Los argumentos usados con la herramienta ls son:

-d	Lista los directorios en lugar de su contenido.
*	Se usa como argumento todos los ficheros y directorios que se encuentren en el directorio de trabajo actual.

El argumento usado con la herramienta grep es:

^copiaSeguridad_	Muestra como resultado aquellas líneas que comienzan con el patrón "copiaSeguridad_".

5. Compruebar que las sumas de comprobación almacenadas en el servidor se corresponden con el contenido de las copias de seguridad mediante el proceso donde se recalculará la suma de comprobación y se comparará con la almacenada. De esta manera se valida que no haya sido modificada. Para ello se hará uso de las herramientas ssh, cd, tar, md5sum y cat.

   Este paso se divide en otros tres pasos menores:

   a. Recalcular la suma de comprobación de los directorios que contienen las copias de seguridad. Para ello se hará uso de las herramientas ssh, tar y md5sum.

```
 ssh obatchmode=yes -obatchmode=yes -oPort=2222 -i rutaClavePublica
usuarioBackup@servidor01 " cd /backup/Rsync/; tar -cf -
rutaDirectorio | md5sum" > /tmp/checksumCalculadoServidor
```

A continuación, se detallarán los diferentes argumentos usado con cada herramienta.

Los argumentos usados con la herramienta ssh son los mismo que se usa en el paso 4 a excepción del siguiente:

/tmp/checksumCalculadoServidor	Indica el archivo donde se almacenará la salida estándar del script.

El argumento usado con la herramienta cd es:

/backup/Rsync/	Indica la ruta del directorio donde se quiere cambiar el directorio de trabajo

Los argumentos usados con la herramienta tar son:

-c	Crea un nuevo fichero comprimido.
-f -	Indica que se vuelque el contenido sobre el fichero del nuevo fichero comprimido en la salida estándar en vez de sobre un fichero concreto.
rutaDirectorio	Indica la ruta del directorio al cual se le va a calcular la suma de comprobación.

b. Consultarar la suma de comprobación de los directorios que contienen las copias de seguridad que se encuentran almacenados en el servidor. Para ello se hará uso de las herramientas ssh y cat[38].

```
 ssh -obatchmode=yes -oPort=2222 -i rutaClavePublica
usuarioBackup@servidor01 " cat archivoChecksum" >
/tmp/checksumGuardadoServidor
```

A continuación, se detallarán los diferentes argumentos usado con cada herramienta.

Los argumentos usados con la herramienta ssh son los mismo que se usan en el paso 5.b. El único cambio es el nombre del fichero donde almacenar la salida estándar; ahora se usa /tmp/checksumGuardadoServidor.

El argumento usado con la herramienta cat es:

archivoChecksum	Indica la ruta absoluta del fichero que contiene la suma de comprobación para rutaDirectorio.

c. Comparar las dos sumas de comprobación para verificar que no se modificó la copia de seguridad. Para ello se hará uso de la herramienta diff.

```
 diff -q /tmp/checksumCalculadoServidor
/tmp/checksumGuardadoServidor
```

Los argumentos usados son:

-q	Muestra un mensaje informando únicamente si los ficheros difieren en su contenido.
/tmp/checksumCalculadoServidor	Indica el fichero que contiene la suma de comprobación calculada.
/tmp/checksumGuardadoServidor	Indica el fichero que contiene la suma de comprobación. Nos servirá como punto de referencia para saber si la copia de seguridad fue modificada.

Si se detectara que ha sido modificado la copia de seguridad en el servidor, se podrá comprobar si en el cliente existe esa misma copia de seguridad y si esta última ha sido modificada o no.

Para ello bastaría con realizar los pasos 5 y 6 especificados en la sección del servidor. Si se comprobara que en el cliente se encuentra la copia de seguridad y no ha sido modificada, se puede copiar en el servidor.

Este paso será necesario repetirlo para todo el resultado obtenido en el paso 4.

2.4.3 Líneas de continuación

Las posibles líneas de continuación podrían ser las siguientes:

- Buscar el procedimiento mediante el cual se puedan copiar propiedades como la inmutabilidad de los archivos.

6. Buscar un método para el cálculo de la suma de comprobación que consuma menos tiempo y recurso que el uso de las herramientas tar y md5sum.

- Idear un procedimiento en el que el número de veces que se calcula la suma de comprobación para comprobar la integridad de las copias de seguridad se reduzca.

2.5 Plan de copias de seguridad

En esta sección se detallará el plan para realizar las copias de seguridad de los servidores del laboratorio de Telemática. Se centrará en cómo plantear una nueva distribución de las unidades de almacenamiento de los servidores y de proporcionar un programa lleve a cabo dichas copias de seguridad.En este plan de copias seguridad se hará uso de todas las subsecciones anteriores.

2.5.1 Distribución de las unidades de almacenamiento

A continuación, se detallará una nueva distribución de las unidades de almacenamiento que habrá en cada servidor. En esta distribución se hará uso de LVM y RAID software. Cada servidor dispondrá de cuatro discos, dos de estado sólido de 1Tb de capacidad y dos de estado sólido de 2Tb de capacidad.

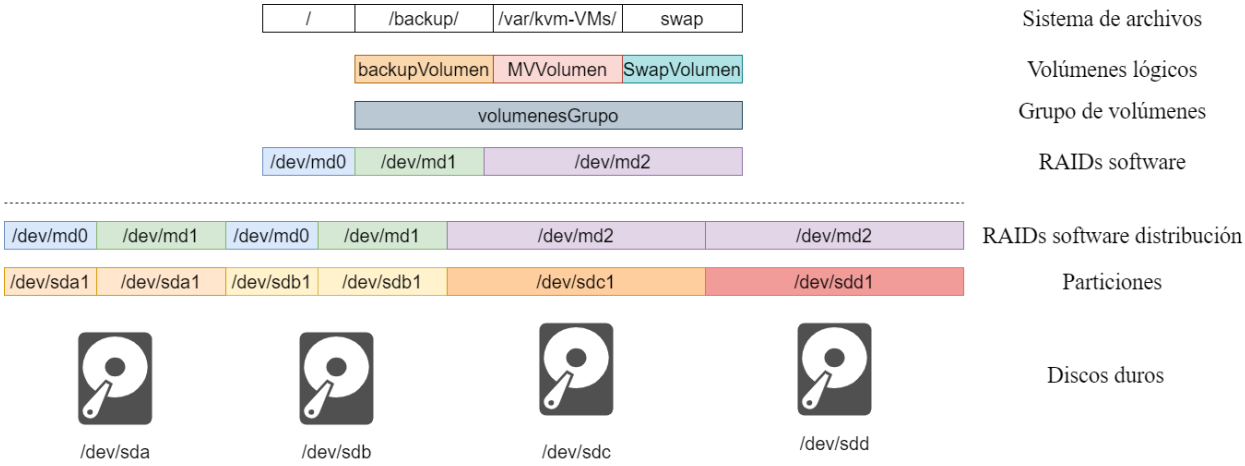

Ilustración 2-18 Diagrama nueva distribución de las unidades de almacenamiento

Los discos se identificarán de la siguiente manera:

- Los de 1Tb se identificará como /dev/sda y /dev/sdb.
- Los de 2Tb se identificará como /dev/sdc y /dev/sdc.

Las particiones que se crearán son las siguientes:

- /dev/sda1 y /dev/sdb1 de 20 Gb.
- /dev/sda2 y /dev/sdb2 con el resto de la capacidad disponible de los discos /dev/sda y /dev/sdb.
- /dev/sdc1 y /dev/sdd1 con toda la capacidad de los discos /dev/sdc y /dev/sdd.

Se crearán tres RAIDs software de nivel 1:

- Las particiones /dev/sda1 y /dev/sdb1 formarán el RAID /dev/md0.
- Las particiones /dev/sda2 y /dev/sdb2 formarán el RAID /dev/md1.
- Las particiones /dev/sdc1 y /dev/sdd1 formarán el RAID /dev/md2.

Se creará un grupo de volúmenes que se nombrará como volumenesGrupo y cuyos volúmenes físicos serán los RAIDs /dev/md1 y /dev/md2. De este grupo se crearán los siguientes volúmenes lógicos:

- Volumen backupVolumen al cual se le asignará 1,5 Tb que se destinará para almacenar las copias de seguridad.
- Volumen SwapVolumen al cual se le asignará 32Gb que se destinará a una partición swap.
- Volumen MVVolumen al cual se le asignará 1.1Tb que se destinará para alojar los archivos qcow2 de las máquinas virtuales KVM en las que se encuentran las aplicaciones y las herramientas usadas del laboratorio.

Por último, el RAID /dev/md0 se destinará a la instalación del sistema operativo para el anfitrión de la máquina virtual KVM. El motivo es que actualmente el sistema LVM no permite la instalación de la partición /boot sobre un volumen lógico, ya que el gestor de arranque GRUB no puede leerlo.

2.5.2 Programa para la realización de las copias de seguridad

A continuación, se detallará el programa a seguir para realizar las copias de seguridad. Se detallará que sobre elementos se harán las copias de seguridad, con que periodicidad, cuantas copias se almacenará y que herramienta se hará uso para ello.

Antes de comenzar se mostrarán los diferentes ficheros qcow2 que posee la máquina virtual KVM. Estos ficheros representan los discos de la máquina.

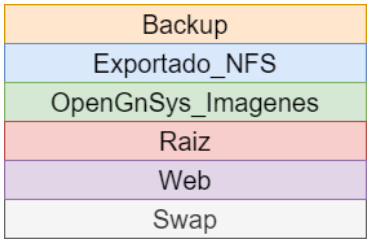

Ilustración 2-19 Fichero qcwo2 que componen la máquina virtual KVM

También se presenta un diagrama donde puede verse los servidores existentes del laboratorio, junto con sus nombres y con las máquinas virtuales que alojan.

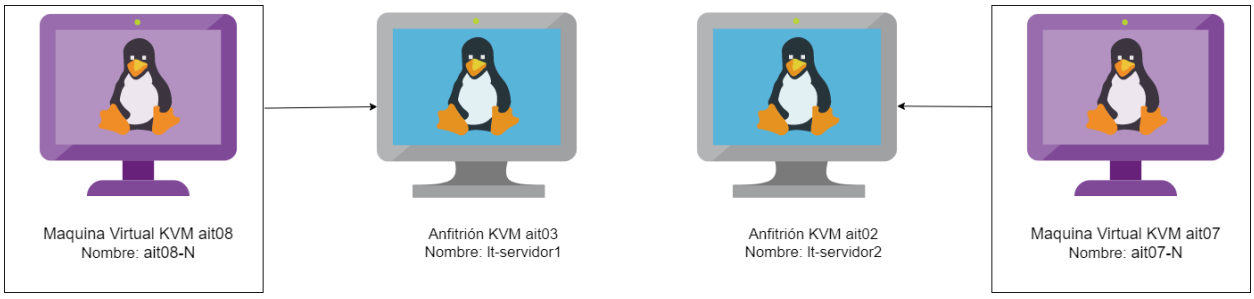

Ilustración 2-20 Diagrama de servidores del laboratorio

Las copias de seguridad que se realizará y almacenará en el lt-servidor1.

Tabla 1 Copias de seguridad en anfitrión

Elemento	Periodicidad	Histórico	Como realizar	Tamaño estimado	Carpeta para almacenar
Sistema operativo sin MV ni copias de seguridad *	Trimestral	Anual	Copia del RAID /dev/md0 con SystemRescue Cd	40Gb	/backup/copiaSergurida_HOST_X siendo X un número entre 1 y 4
Fichero XML de la máquina virtual y el fichero qcow2 Raiz	Trimestral	Anual	Se usará la herramienta rsync	40Gb	/backup/copiaSergurida_MV_xml_raiz_X siendo X un número entre 1 y 4
Todos los ficheros qcwo2 excepto Raiz y OpenGnSys_Imagenes	Mensual	Mensual	Se usará la herramienta rsync	400Gb	/backup/copiaSergurida_MV_no_raiz_opengnsys
Fichero qcow2 OpenGnSys_Imagenes	Mensual	Mensual	Se usará la herramienta rsync	700Gb	/backup/copiaSergurida_MV_opengnsys
Configuraciones del anfitrión	Diaria	Bianual	Se usará el procedimiento de git como servidor	50Gb	/backup/copiaSergurida_Repositorio_HOST
Configuraciones de la máquina virtual.	Diaria	Bianual	Se usará el procedimiento de git como cliente	50Gb	/backup/copiaSergurida_Repositorio_MV

*Se aprovechará para realizar la copia de todos los elementos necesarios para recrear los RAIDs software y los grupos de volúmenes LVM que componen el sistema excepto las particiones de datos que se clonan mediante el uso de fsarchiver.

Las copias de seguridad que se realizarán y almacenarán en el ait08-N.

Tabla 2 Copias de seguridad en la máquina virtual

Elemento	Periodicidad	Histórico	Como realizar	Tamaño estimado	Carpeta para almacenar
Configuraciones de la máquina virtual.	Diaria	Bianual	Se usará el procedimiento de git como servidor	50Gb	/backup/copiaSergurida_Repositorio_MV

Se hará uso de la herramienta cron para realizar la programación de las copias de seguridad. Se harán de todas menos de aquellas que impliquen el uso de SystemRescueCd, ya que es necesario realizarla manualmente porque requiere del reinicio del sistema.

Para finalizar, se usará el procedimiento explicado en la subsección Sincronización de copias de seguridad mediante rsync para crear una réplica de las copias de seguridad del servidor lt-servidor1 en lt-servidor2, donde lt-servidor1 actuará como servidor y lt-servidor2 como cliente.

3 ALTA DISPONIBILIDAD

En este capítulo se abordará algunas soluciones que permitirán mantener en funcionamiento del laboratorio en caso de fallo de algún servicio o fallo de un servidor completamente. Entre las soluciones que se hará el uso de herramientas que faciliten la administración de las máquinas virtuales que proporcionan servicios al Laboratorio de Telemática y los servidores que la alojan.

3.1 Configuración de iLO

En esta sección se explicará el proceso de configuración del iLO (Integrated Lights-Out)[39] que posee los algunos servidores HP.

Esta interfaz es un procesador que incluye algunos servidores ProLiant de la marca HP y que permite la gestión remota de los servidores que lo incluye. Este procesador es independiente del procesador instalado en el servidor y hace uso de recurso independiente para su funcionamiento.

Además, posee conexión directa una interfaz de Ethernet dedicada y también puede configurarse para que haga uso de una de las conexiones Ethernet que usa el sistema, pero es algo desaconsejado.

De esta interfaz existen varias generaciones[40]:

Generación	Generación de servidores ProLiant
iLO 5	Generación 10
iLO 4	Generación 9 y 8
iLO 3	Generación 7

Existen otras dos generaciones, pero HP dejo de darle soporte por pertenecen a generaciones de servidores ProLiant más antigua.

Entre las ventajas más interesante que nos permite el uso de la interfaz son:

- visualización de estado de la salud del dispositivo
- Botones de encendido virtuales
- Control remoto completa del servidor
- Dispositivos virtuales integrado en el control remoto
- Web de gestión
- Gestión remota mediante ssh

3.1.1 Situación actual y problemática

En el laboratorio actual no se dispone un procedimiento que permita realizar un apagado o reinicio forzoso de manera remota que permita salir de un bloqueo generalizado o la aplicación de una configuración de red incorrecta de los sistemas operativos de los servidores.

Esto implica que si se produce un bloqueo generalizado o se deshabilitan las interfaces de red sea necesario realizar el apagado o reinicio de forma presencial.

Sin embargo, en el laboratorio se posee dos servidores ProLiant ML310e Gen8 v2, los cuales poseen una interfaz

iLO 4, esta una vez configurada permitirá poder realizar un realizar apagado o reinicio desde su página web de gestión de manera remota.

Además, este permite el control de servidor mediante una consola remota que observar el monitor y controlar tanto el teclado como el ratón independiente de si nos encontramos en el arranque del servidor o en el sistema operativo, aunque este se ve limitado solo hasta el arranque del sistema operativo sino se posee de una licencia iLO Advanced.

A continuación, se detallarán los procedimientos que se encuentra en el manual de usuario de iLO 4 y que se son básico para poder ser usada para el encendido y apagado remoto:

1. Configuración de la interfaz de red de acceso a la interfaz de iLO.
2. Actualización del firmware de iLO vía interfaz web.
3. Configuración de puerto y usuarios para el acceso vía interfaz web.

En el ANEXO E: Configuración iLO se detallará las ventajas interesantes que ofrece la adquisición de una licencia y como configurarla.

3.1.2 Configuración de la interfaz de red

En este proceso se realizará una configuración mediante el uso de interfaz de Ethernet dedicada a iLO y se le asignará una dirección IP publica fija. Este es una de la posible manera detallada en el manual de usuario de iLO 4 y esta requiere que se realice durante el inicio del servidor.

En el siguiente diagrama obtenido del manual del servidor ProLiant ML310e Gen8 v2[41] podemos observar en el punto 9 el puerto Ethernet dedicado a iLO. Este puerto es necesario que se conecte a la red externa del laboratorio para su posterior uso.

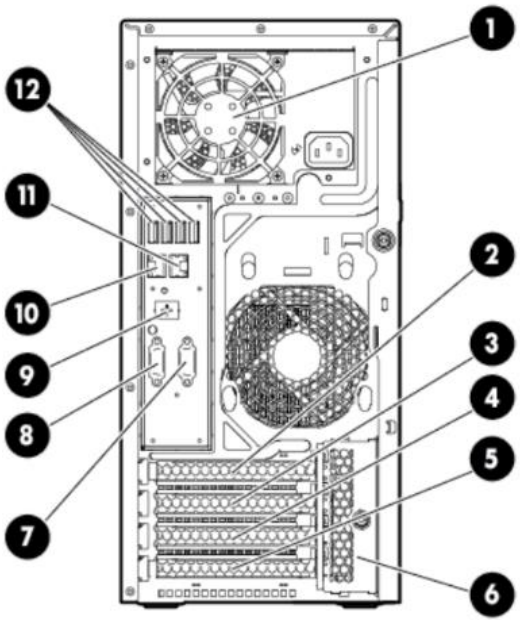

Ilustración 3-1 Diagrama de la parte trasera de ProLiant ML310e Gen8 v2

Los pasos que se deben seguir son:

1. Apagar completamente el servidor en el cual se va a realizar la configuración.
2. Pulsar [F8] cuando aparezca el mensaje: iLO Advanced press [F8] to configure.

Ilustración 3-2 Pantalla de arranque de BIOS de ProLiant ML310e Gen8 v2

3. Navegar mediante las teclas de dirección por los siguientes menús hasta llegar a la entrada NIC and TCP/IP y pulsar [Intro].

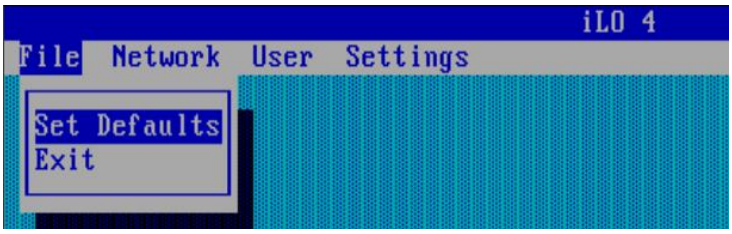

Ilustración 3-3 Menú de configuración la interfaz iLO

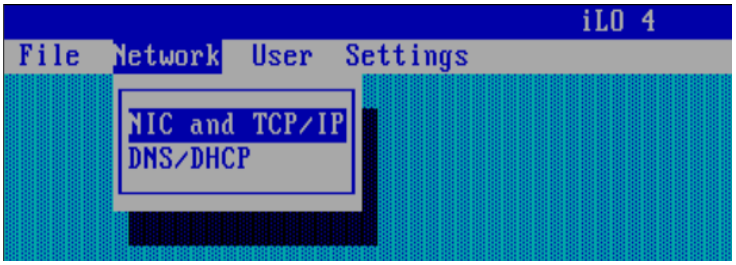

Ilustración 3-4 Menú de configuración de red de la interfaz iLO

4. Navegar mediante las teclas de dirección hasta el campo Network Interface Adapter donde se debe pulsar [Barra espaciadora] hasta que el valor sea ON.

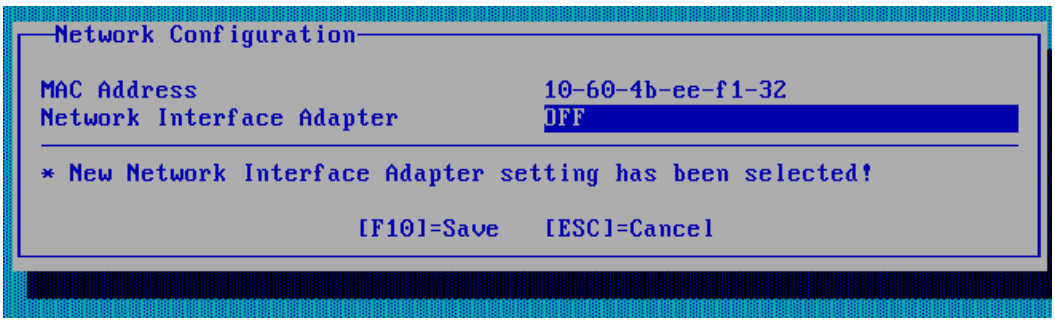

Ilustración 3-5 Funciones de red de la interfaz iLO desactivado

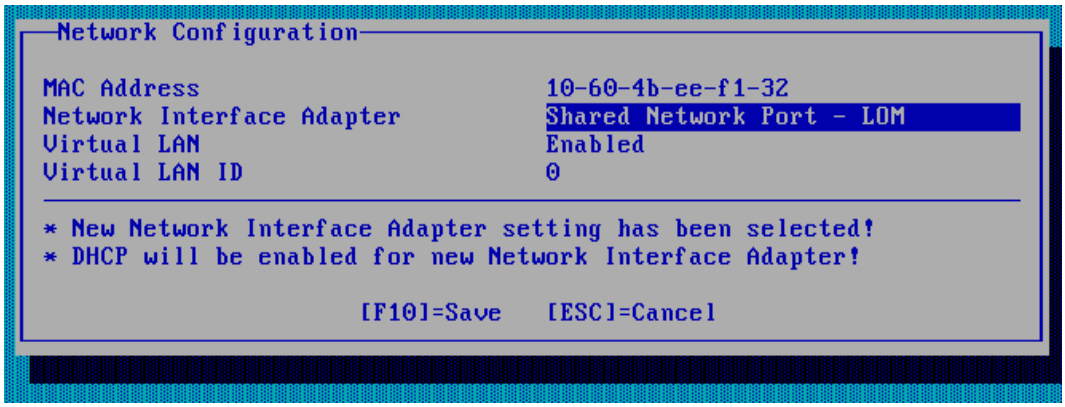

Ilustración 3-6 Funciones de red de la interfaz iLO activa y compartida con el Sistema Operativo

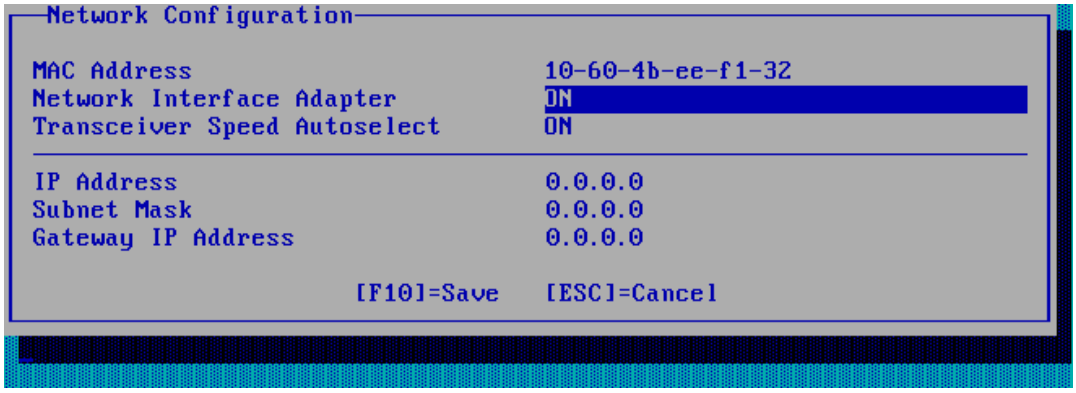

Ilustración 3-7 Funciones de red de la interfaz iLO activa e independiente

5. Navegar mediante las teclas de dirección por los campos IP Address, Subnet Mask y Gateway IP Address mientras se introducen lo datos para configurar la IP publica que se quiera asignar al iLO.

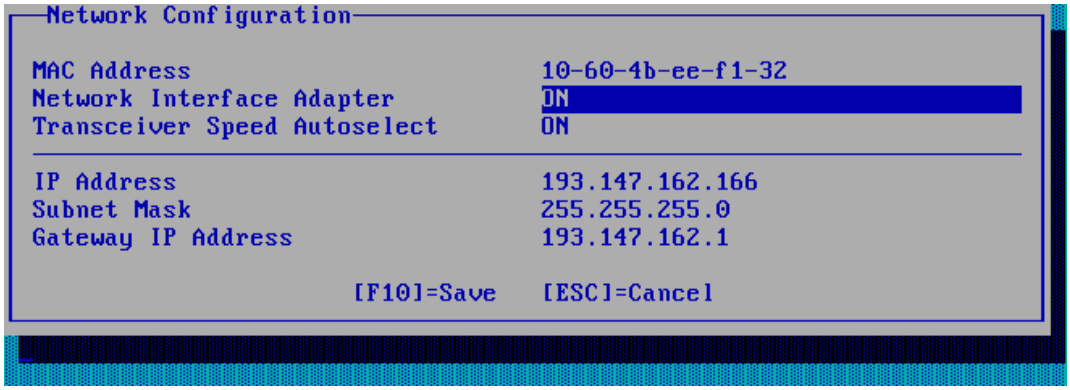

Ilustración 3-8 Funciones de red de la interfaz iLO activa e independiente con valores asignados

6. Salir de la configuración de la interfaz para ello pulsar [F10] y con ello guardar la configuración realizar.
7. Navegar mediante las teclas de dirección por los siguientes menús hasta llegar a la entrada DNS/DHCP y pulsar [Enter].

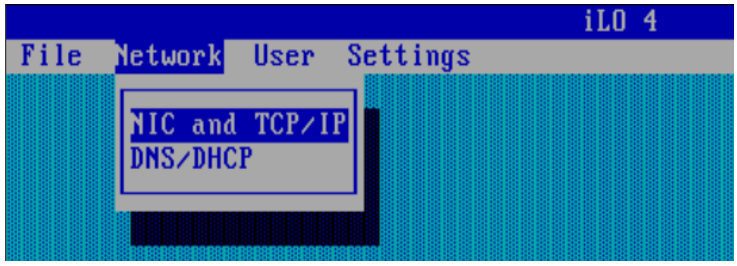

Ilustración 3-9 Menú de configuración de red de la interfaz iLO

8. Navegar mediante las teclas de dirección hasta el campo DHCP Enable donde se debe pulsar [Barra espaciadora] hasta que el valor sea OFF.

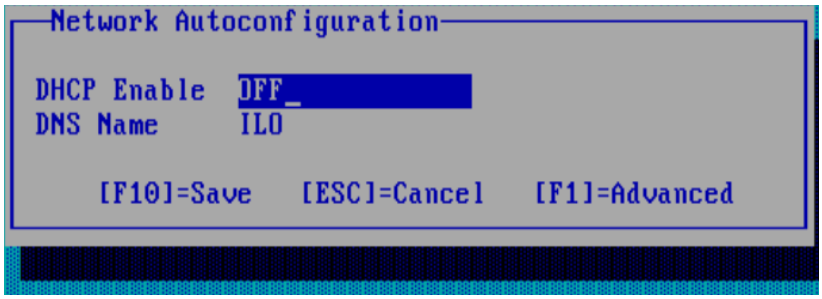

Ilustración 3-10 Funciones de DHCP de la interfaz iLO destivada

9. Salir de la configuración de la interfaz para ello pulsar [F10] y con ello guardar la configuración realizar.
10. Navegar mediante las teclas de dirección hasta el campo Exit donde se debe pulsar [Barra espaciadora].

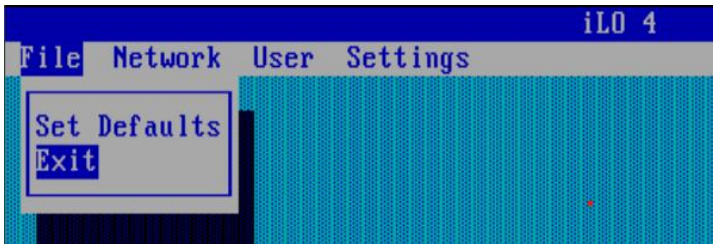

Ilustración 3-11 Menú de configuración la interfaz iLO antes de salir

Tras realizar los pasos el servidor se reiniciarla y ya será accesible la web de gestión desde el navegador. En la configuración mostrada la URL sería https://193.147.162.166/

Por defecto existe configurado un usuario y contraseña con el rol de administrador, estos están detallado en una pegatina que con el título iLO Default Network Setting.

3.1.3 Actualización de firmware

En este proceso hará uso de la interfaz web para realizarlo, pero en el manual de usuario de iLO 4 se detallan otros posibles métodos. Los menús y pantalla de la interfaz web pueden varias en función de la versión de firmware.

Los pasos que se deben seguir son:

1. Descargar el binario para la interfaz iLO para ello se accederá a la web https://support.hpe.com/, buscar por la palabra clave iLO 4, presionar la pestaña controlador y software y seleccionar Firmware en Tipo de software.

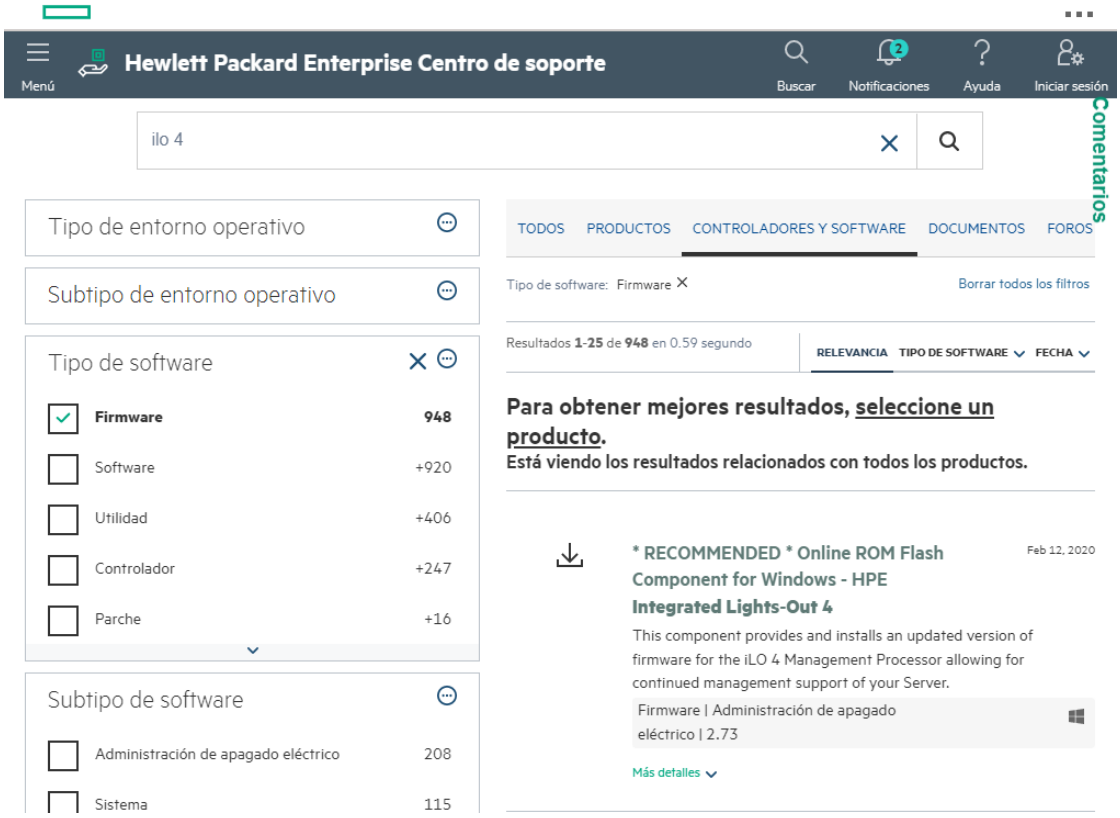

Ilustración 3-12 Página de descarga firmware para iLO

Seleccionamos la última versión dependiendo del sistema operativos se vaya a utilizar para acceder a la web

2. Descomprimir el paquete descargado para el archivo binario del firmware.

Si hacemos uso de Windows los pasos para descomprimirlo son:

a) Descargar el fichero cpXXXXXX.exe

Ilustración 3-13 Página de descarga firmware concreto para iLO en formato Windows

b) Descomprimir el contenido del fichero cpXXXXXX.exe pulsando el botón derecho sobre el ratón y seleccionando la opción extraer en cpXXXXXX

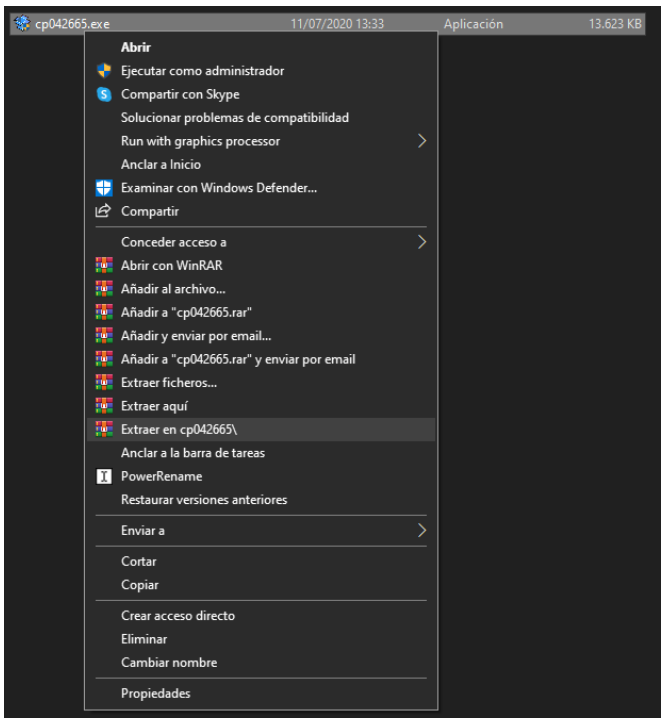

Ilustración 3-14 Menú para la descompresión del firmware de iLO en Windows

c) Obtener del directorio cpXXXXXX se encontrará el archivo binario ilo4_xxx.bin que será el fichero necesario para realizar la actualización.

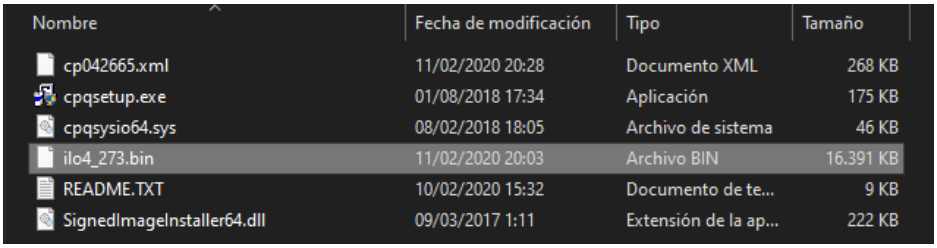

Ilustración 3-15 Ubicación del fichero firmware de iLO en Windows

Si hacemos uso de Linux los pasos para descomprimirlo son:

a) Descargar el fichero CPXXXXXX.scexe

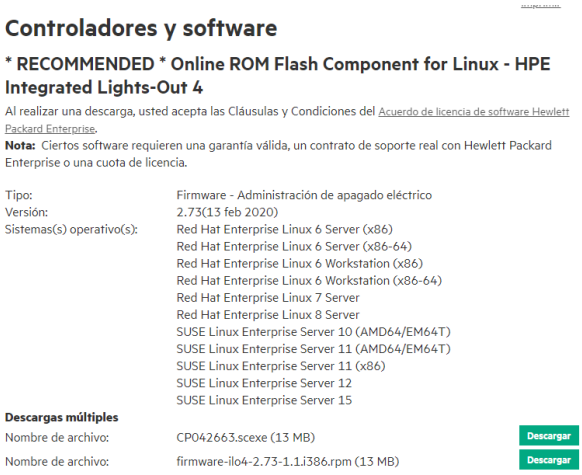

Ilustración 3-16 Página de descarga firmware concreto para iLO en formato Linux

b) Ejecutar la siguiente secuencia de comando para obtener el binario

```
sh CP0*.scexe --unpack=directory
```

c) Obtener del directorio directory se encontrará el archivo binario ilo4_xxx.bin que será el fichero necesario para realizar la actualización.

3. Acceder a la página de web de iLO y una vez autentificado pulsar en la entrada del menú lateral Firmware en el submenú de Administration.

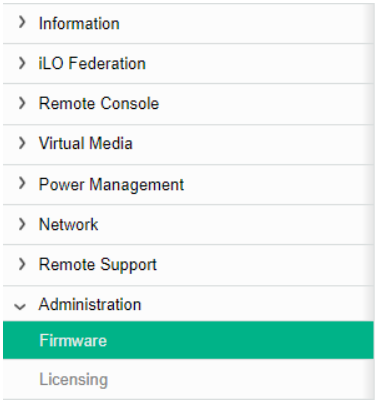

Ilustración 3-17 Ubicación de la página de firmware en el menú lateral de la web de iLO

4. Seleccionar el botón seleccionar fichero y elegir el archivo binario que se obtenido en el paso anterior.

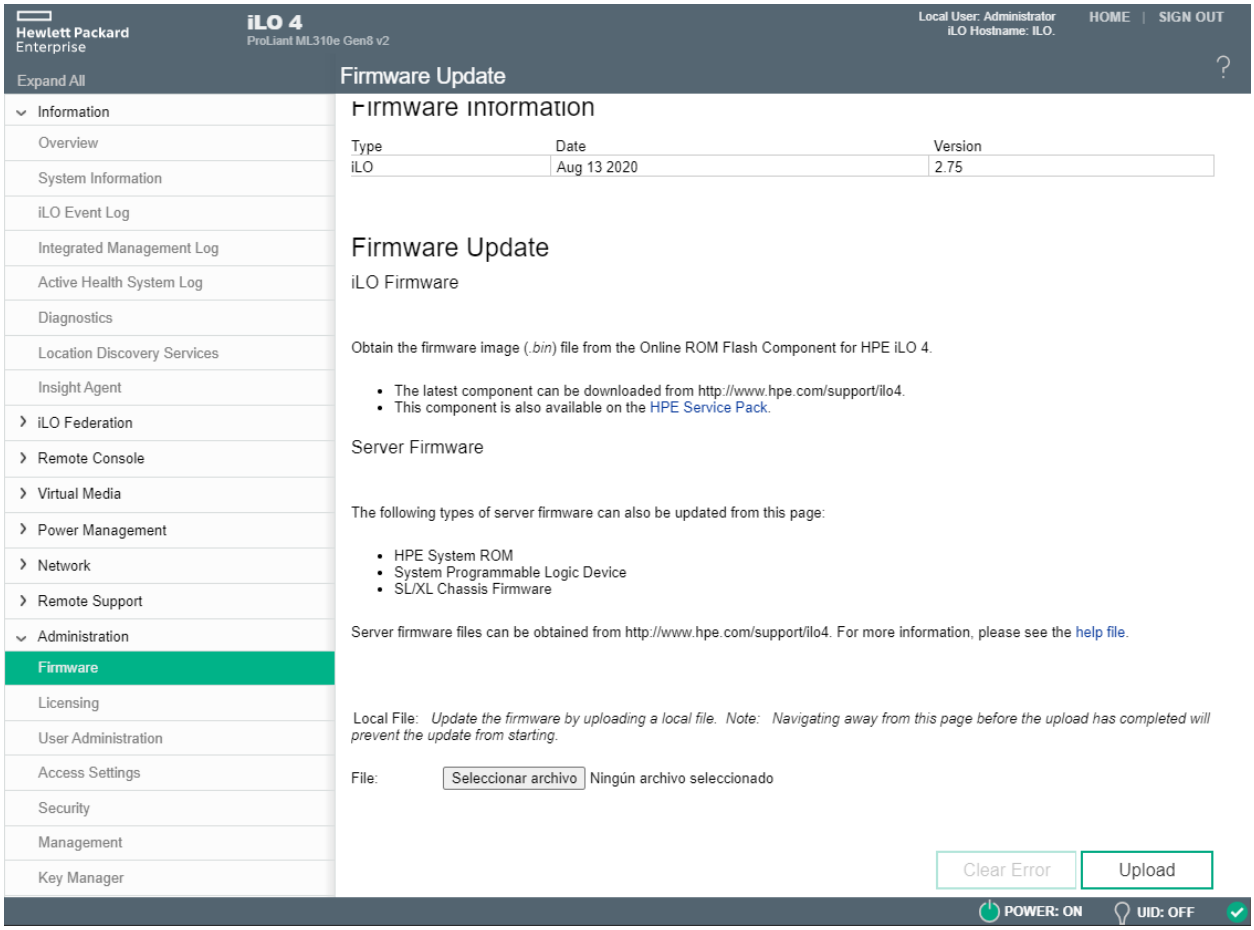

Ilustración 3-18 Página de actualización de firmware de iLO

5. Pulsar el botón Upload y aceptar el popup de advertencia.

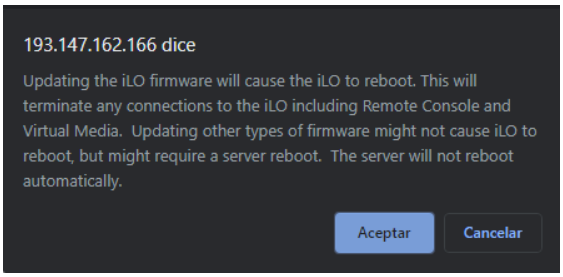

Ilustración 3-19 Popup de confirmación actualización de firmware de iLO

Durante el proceso se pasará por las siguientes fases

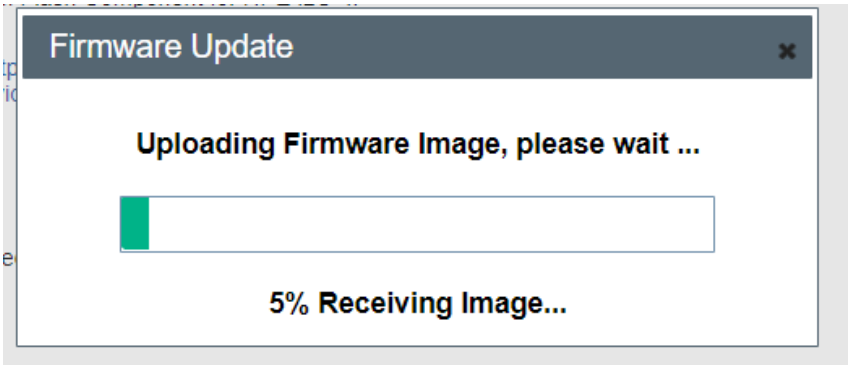

Ilustración 3-20 Proceso de subida del firmware de iLO

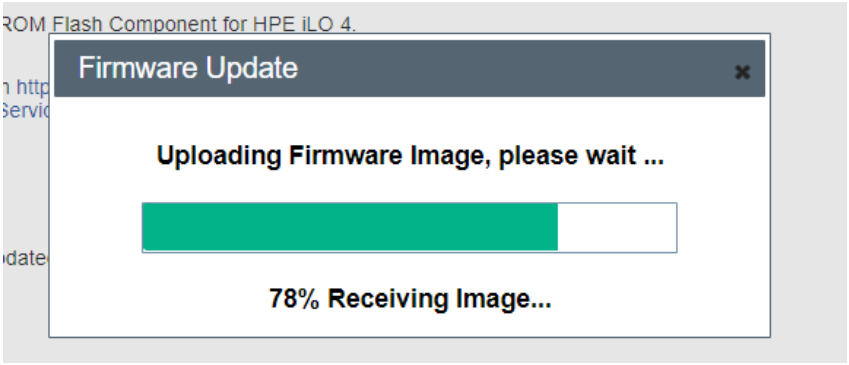

Ilustración 3-21 Proceso de avance subida del firmware de iLO

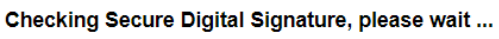

Ilustración 3-22 Inicio actualización del firmware de iLO

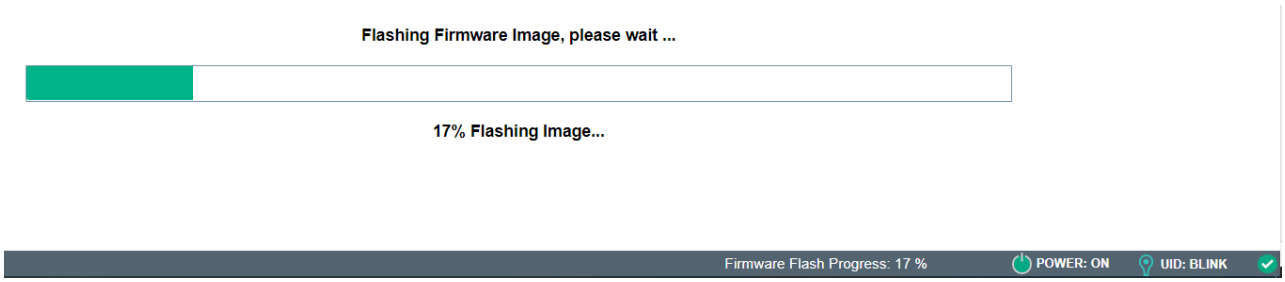

Ilustración 3-23 Progreso actualización del firmware de iLO

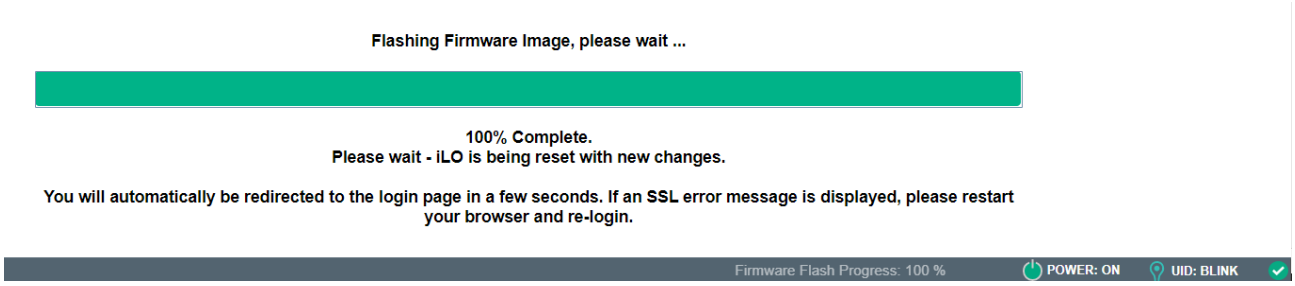

Ilustración 3-24 Fin actualización del firmware de iLO

3.1.4 Configuración de puertos y usuarios

Este proceso hará uso de la interfaz web para realizarlo, pero en el manual de usuario de iLO 4 se detallan otros posibles métodos. Los menús y pantalla de la interfaz web pueden varias en función de la versión de firmware.

Los pasos para la configuración de los puertos que se deben seguir son:

1. Acceder a la página de web de iLO y una vez autentificado pulsar en la entrada del menú lateral Access Settings en el submenú de Administration. Este menú puede variar su apariencia dependiendo de la versión actual.

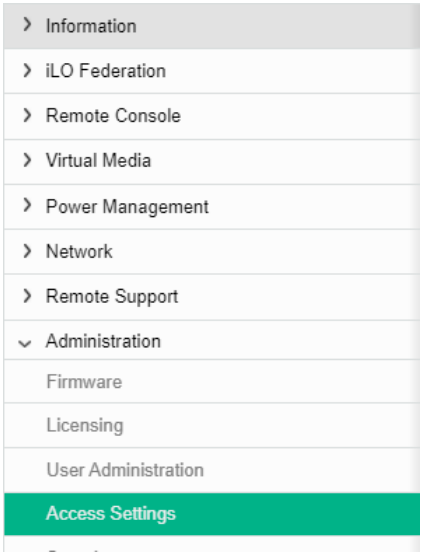

Ilustración 3-25 Ubicación de la página la configuración de puertos en el menú lateral de la web de iLO

2. Seleccionar el número de los puertos. Debido a la funcionalidad de la que se hará uso en el laboratorio se deshabilitará el acceso mediante SSH, SNMP y IPMI/DCMI

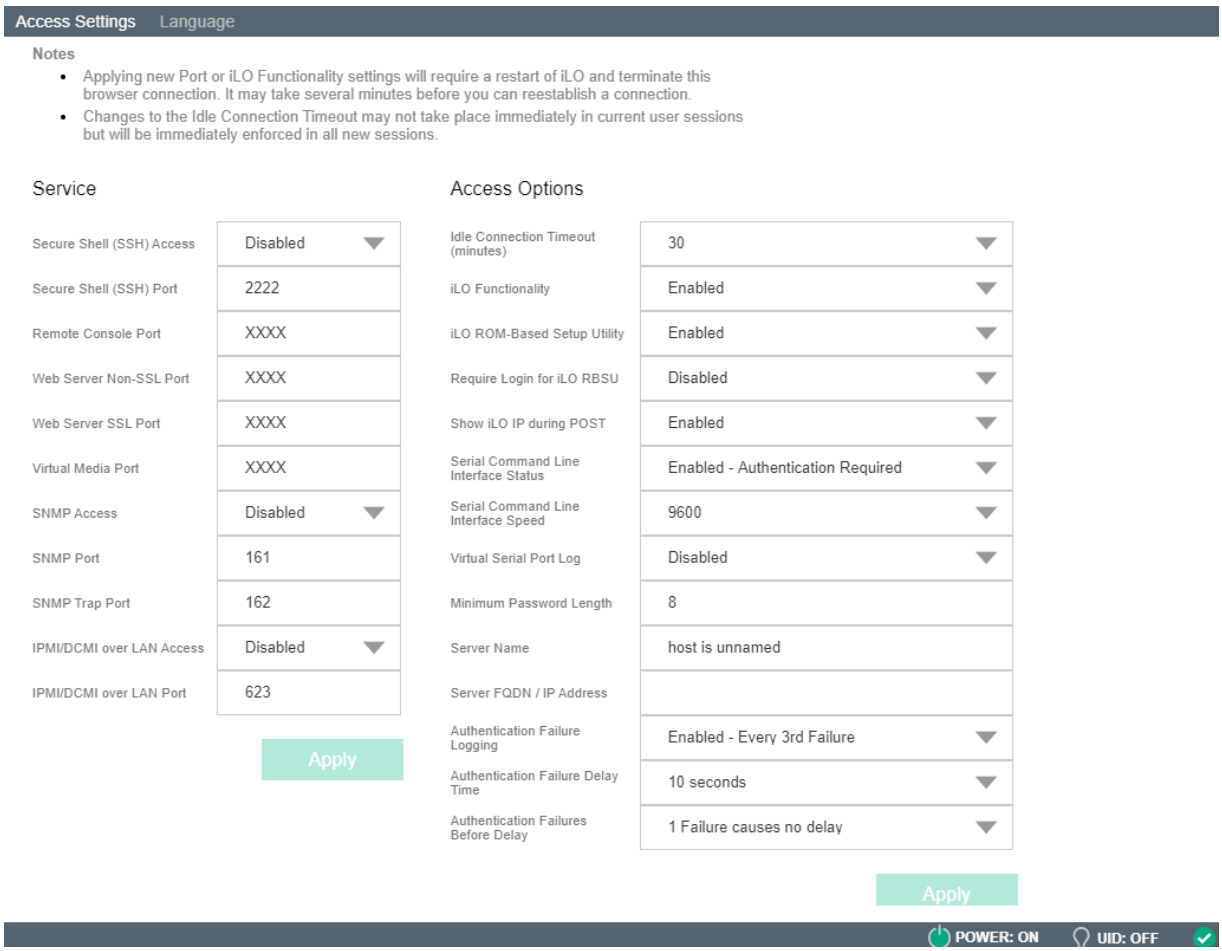

Ilustración 3-26 Página la configuración de puerto de iLO

6. Pulsar el botón Apply y aceptar el popup de advertencia.

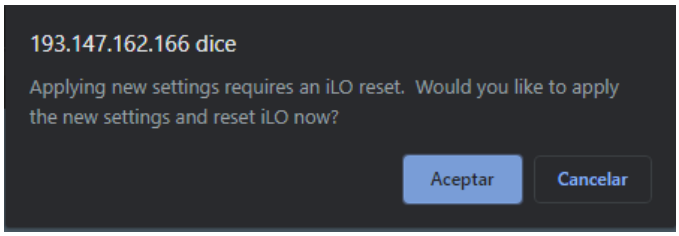

Ilustración 3-27 Popup confirmación para aplicar cambios en la configuración de puertos de iLO

Para la administración de los usuarios se debe acceder a la página de web de iLO y una vez autentificado pulsar en la entrada del menú lateral User Administration en el submenú de Administration.

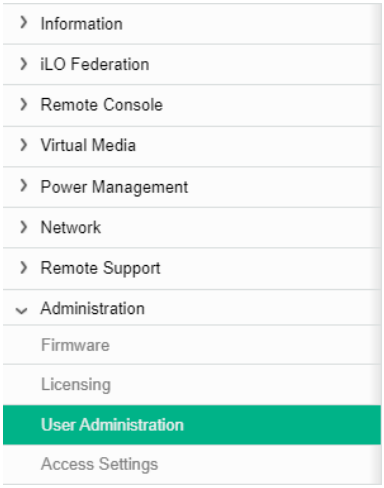

Ilustración 3-28 Ubicación de la página la configuración de usuarios en el menú lateral de la web de iLO

Una vez pulsado se podrá ver la siguiente sección en la pantalla donde se mustra todos los usuarios del sistema y sus permisos.

Ilustración 3-29 Página de usuarios de iLO.

Si se desea crear un usuario habrá que seleccionar el botón New si se quiere crear nuevo usuario y seleccionar el usuario y posteriormente el botón Edit si lo quiere editar.

Tras esto se deberá seleccionar el nombre, usuario, contraseña y permiso asociado al usuario que se quiere crear o editar en la siguiente pantalla.

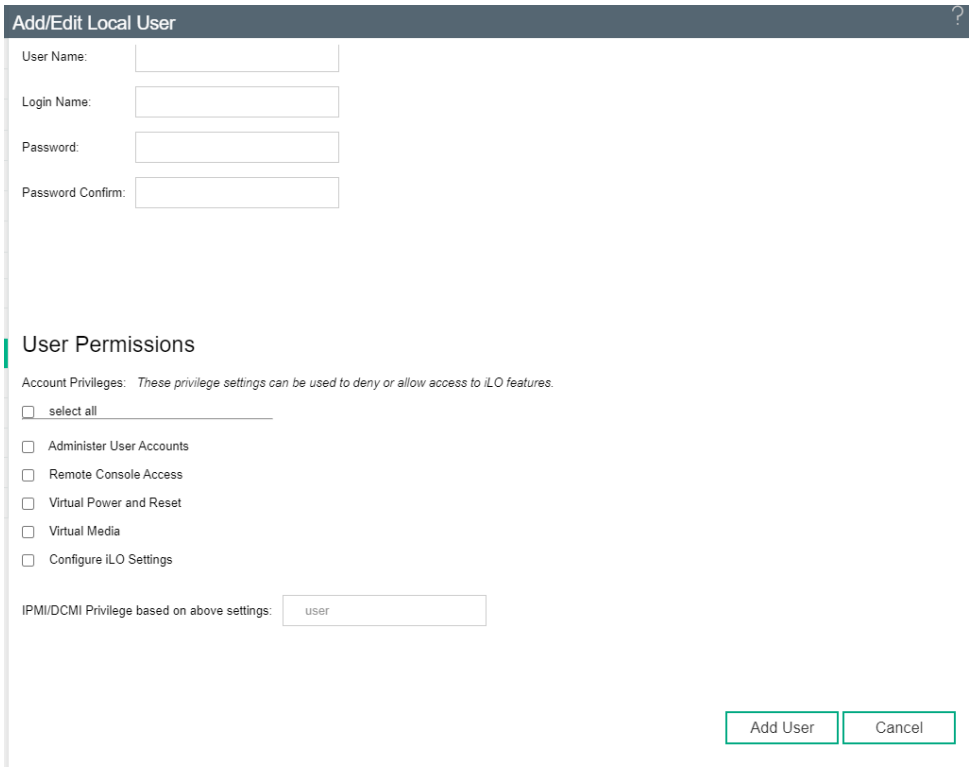

Ilustración 3-30 Creación o actualización de usuarios en iLO

Para finalizar se debe pulsar botón de Add User en el caso de la creación o Update User en el caso de la edición.

Si lo que se quiere es eliminar uno o varios usuarios se deben seleccionar en la pantalla donde se mostra todos los usuarios y pulsar el botón Delete.

Tras esto se deberá aceptar el popup de advertencia que aparecerá para completar la eliminación.

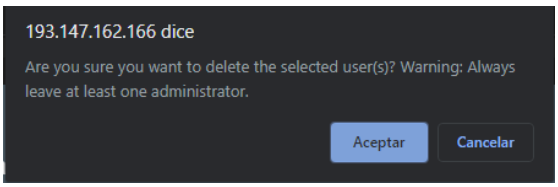

Ilustración 3-31 Popup de confirmación de eliminación de usuario de iLO

3.1.5 Líneas de continuación

En las líneas de continuación se puede valorar la compra de una licencia anual para poder controlar totalmente el servidor en remoto y estudiar la administración mediante la API y/o SSH de iLO.

3.2 Gestión de máquinas virtuales QEMU-KVM

En esta sección se detallará una serie de procedimientos que permiten la administración de máquinas virtuales basada en las tecnologías KVM[42] y QEMU[43].

La tecnología KVM se trata de una funcionalidad del kernel de Linux desde la versión 2.6.20 que permite la virtualización completa del procesador proporcionando el sistema operativo como un hipervisor 1, es decir, un hipervisor sin que el sistema operativo actúe como intermediario proporcionando un mayor rendimiento.

Esta tecnología no puede ser usada sin una herramienta que permita la virtualización del hardware necesario para el correcto de la máquina virtual, para ello se hace uso de la herramienta QEMU. Esta herramienta hace uso de KVM como acelerador de CPU.

3.2.1 Situación actual y problemática

En el laboratorio de Telemática se tiene una máquina virtual en cada servidor basada en las tecnologías QEMU, KVM y administrada mediante la API libvirt[44] donde se almacena y se ejecutan todas las herramientas necesarias para el funcionamiento del laboratorio.

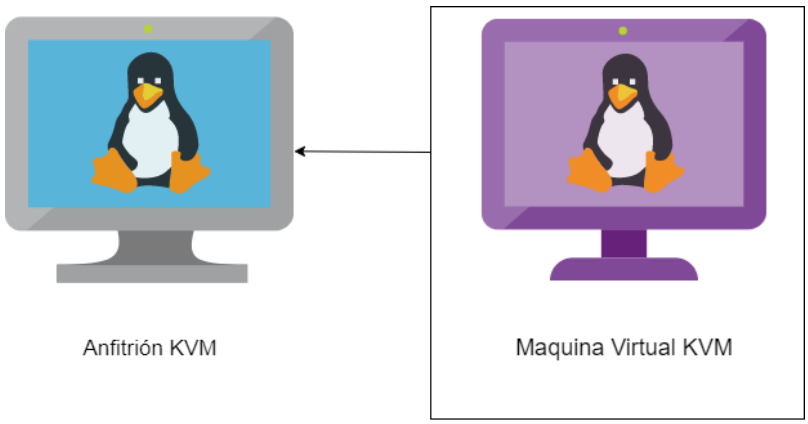

Ilustración 3-32 Escenario para las máquinas virtuales

Para la gestión de las máquinas virtuales se hace uso de las herramientas [45] y virt-viewer[46].

Se quiere proporcionar un método intuitivo al usuario para que el usuario pueda gestionar las máquinas virtuales de manera remoto ya que no es posible el uso de la herramienta virt-manager cuando se acceda mediante ssh. Además, se quiere configurar el acceso remoto a la máquina virtual mediante el protocolo SPICE[47], ya que este permite el control mediante una interfaz gráfica o la conexión remota de USB entre algunas de sus características.

Entre los procesos que se quieren proporcionara son el inicio o apagado de la máquina, exportación o importación de una máquina virtual o la consulta del estado de la máquina virtual.

Como solución se hará uso de la herramienta dialog[48] para la creación de menú que abstraigan al usuario del uso de la herramienta virsh y se crearán procedimiento para permitir la exportación e importación de máquinas virtuales para poder replicar maquina virtuales en el mismo sistema o en otros sistemas.

3.2.2 Menú de gestión de maquina virtuales QEMU-KVM

Se ideado una serie scripts mediante el uso de dialog que permite la ejecución de la herramienta virsh más comunes en la gestión de las máquinas virtuales de una manera más sencilla y sin el usuario tenga que conocer la sintaxis concreta. Estos menús se harán uso a través de tres scripts exportarMVMenu.sh, importarMVMenu.sh y menuVirshCompleto.sh. El principal de ellos es menuVirshCompleto.sh y debe ser ejecutado como root.

Entre las tareas que se han incluido son:

- Listar las máquinas virtuales del sistema realizando el filtrado en función del estado de estas.
- Cambiar el estado de la máquina, permitiendo iniciarla, pararla, suspender o reiniciarla.
- Editar la configuración de la maquina a través del XML asociado a la máquina virtual.
- Renombrar la máquina virtual.
- Acceder a la máquina virtual a través de la consola o mediante virt-viewer[46].
- Habilitar y deshabilitar el autoarranque de la máquina virtual.
- Eliminar la máquina virtual, con o si sus unidades de almacenamiento asociada.
- Exportar o importar máquinas virtuales. Este proceso se detallará en el ANEXO F: Gestión de máquinas virtuales QEMU-KVM.

Para la navegación entre los menús se deberá hacer uso de las teclas dirección, cuando se deba seleccionar un elemento habrá que pulsar la tecla [Barra espaciadora] y para seleccionar entre las opciones de aceptar, salir o atrás habrá que pulsar la tecla [Intro].

En el anexo se detallará el procedimiento usado para realizar la exportación o importación de máquinas virtuales, el uso de los diferentes scripts que componen los menús y la estructura del fichero de configuración de los scripts.

A continuación, se muestra algunas capturas en la que se ve los diferentes menús disponibles.

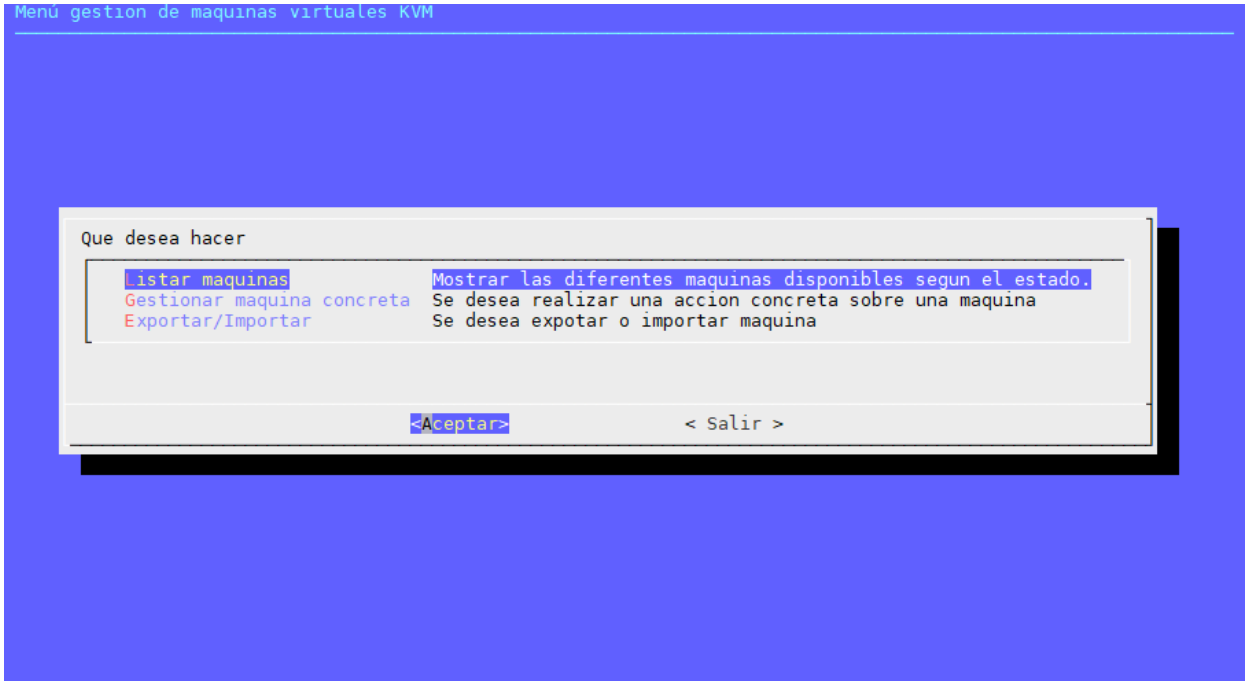

Ilustración 3-33 Menú inicial para administración de máquinas virtuales

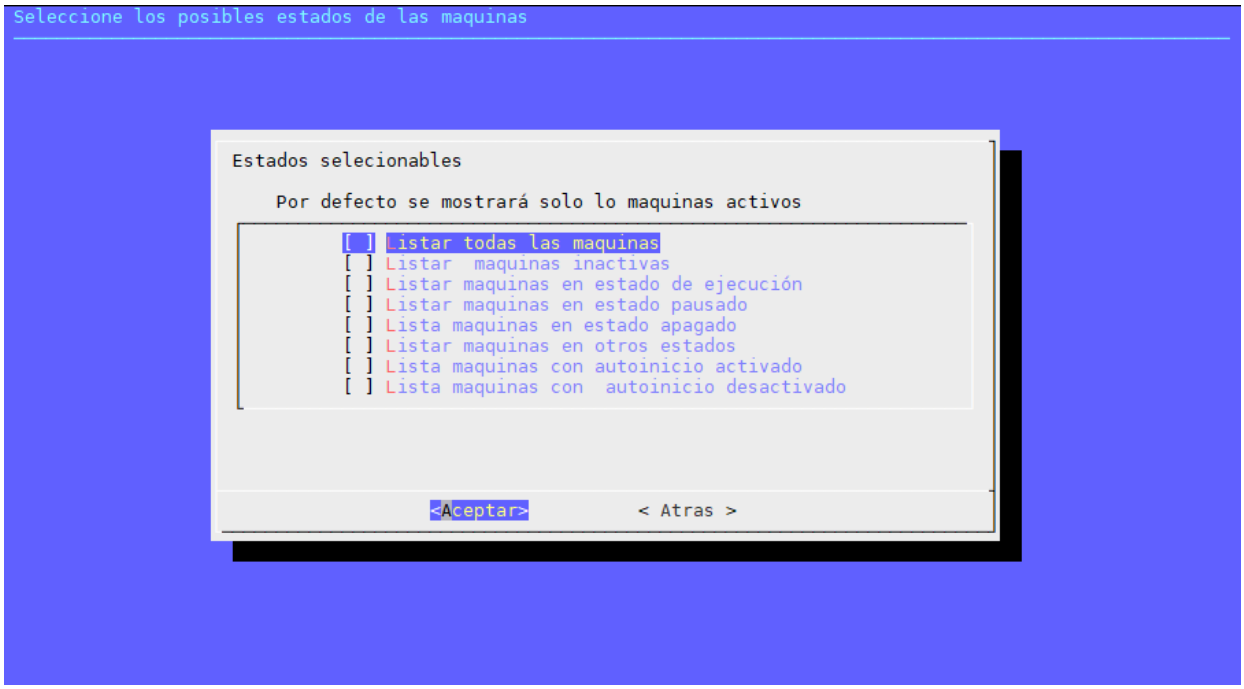

Ilustración 3-34 Menú de estado de las máquinas virtuales

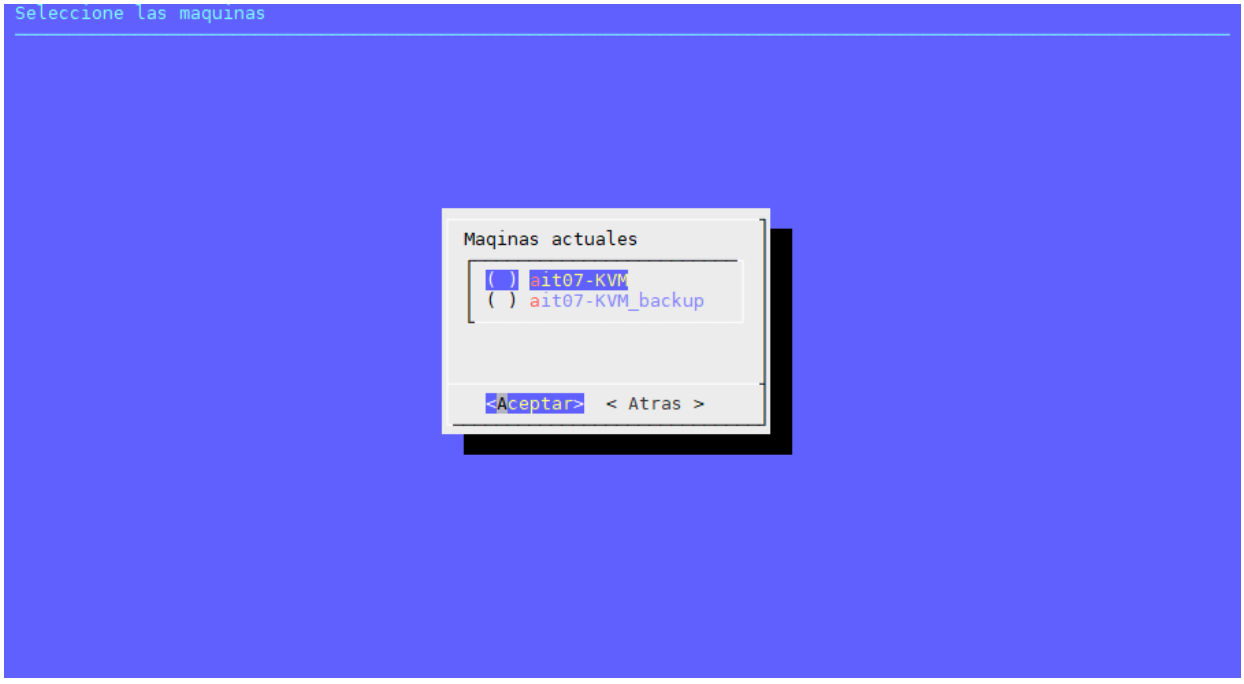

Ilustración 3-35 Selección de máquinas virtuales

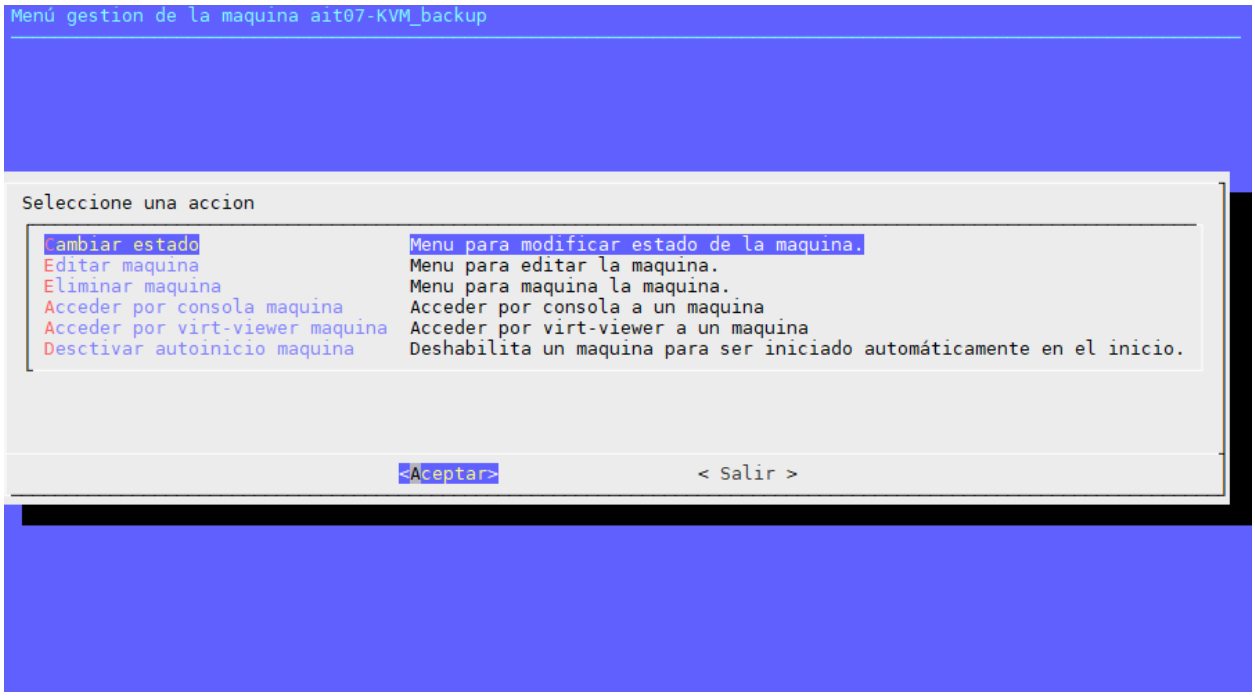

Ilustración 3-36 Menú de gestión de una máquina virtual concreta

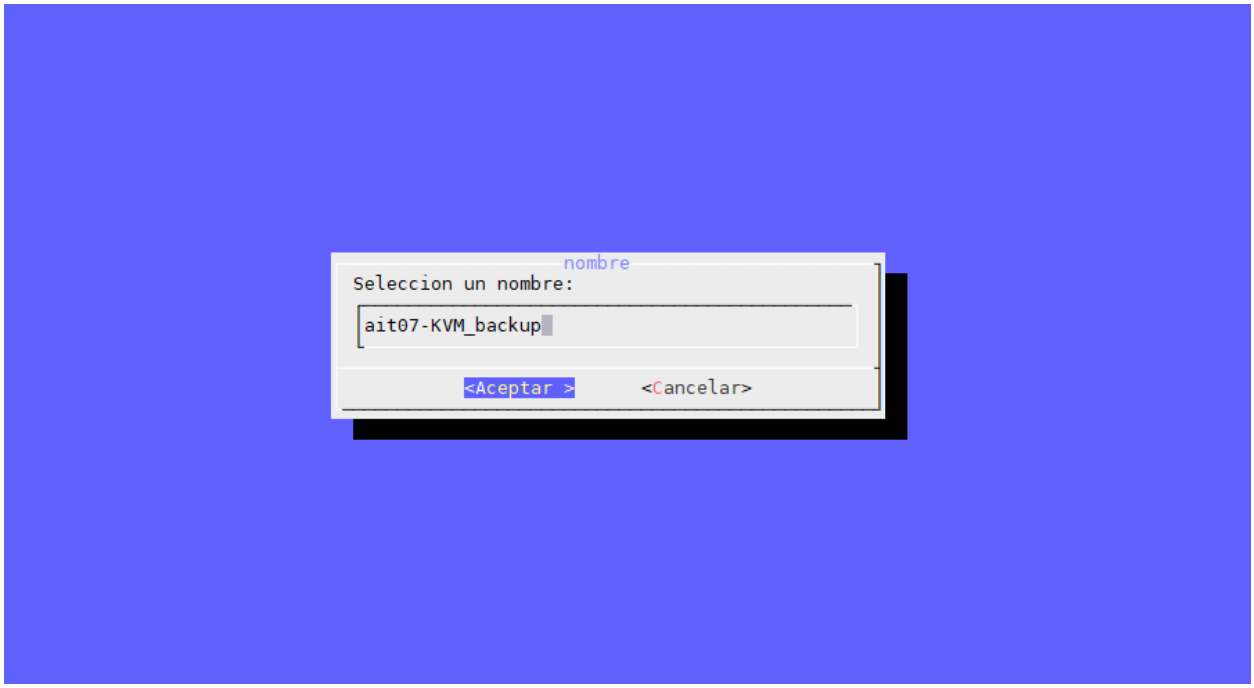

Ilustración 3-37 Cambio del nombre de una máquina virtual

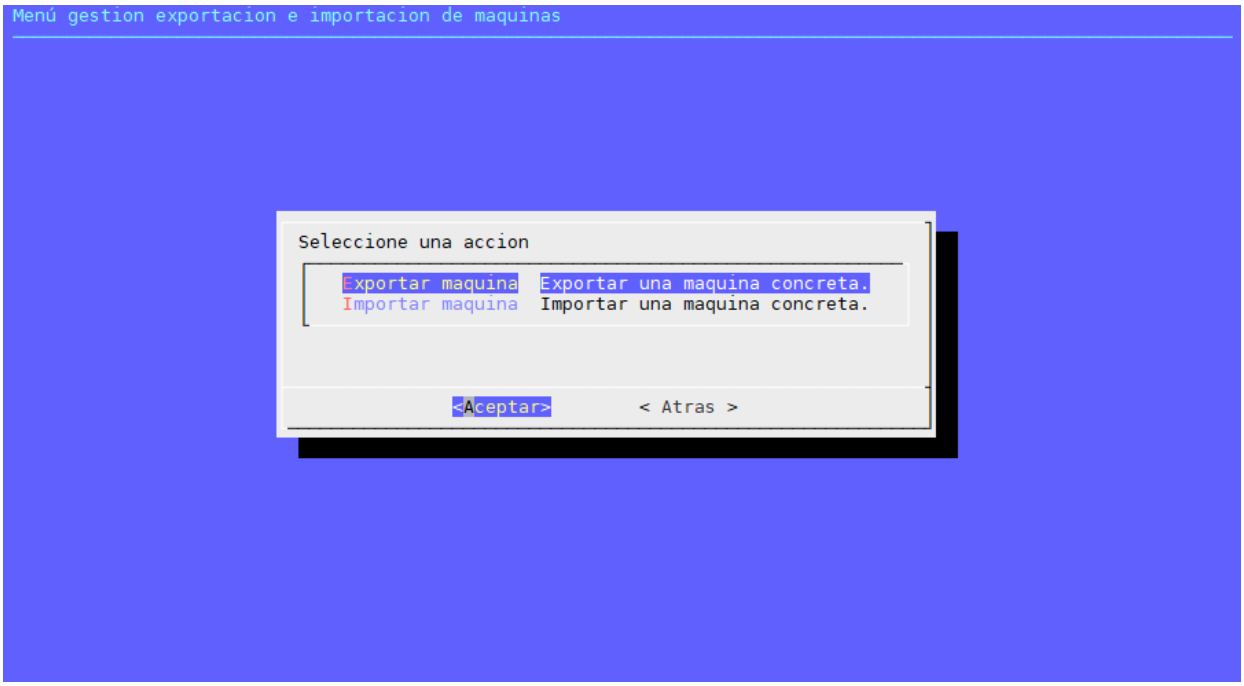

Ilustración 3-38 Menú de importación y exportación de máquinas virtuales

3.2.3 Configuración de protocolo SPICE en máquinas virtuales QEMU-KVM

Las máquinas virtuales usada actualmente en el laboratorio se crean y define mediante el uso la API libvirt, esta define las especificaciones de las máquinas virtuales mediante un archivo XML. En este podemos encontrar la definición de los ficheros que representa unidades de almacenamiento asociada a esta, las interfaces de red que tiene esta y como se relaciona con estas interfaces con las interfaces del host o los recursos como CPU o RAM del anfitrión que se destinará a esta.

Para el resto de los elementos que se pueden definir en el fichero XML se puede consultar en la documentación del dominio XML de libvirt[49].

En esta subsección se detallará como configurar a través de libvirt el uso de SPICE para el control grafico de la máquina, permitir la conexión remota de USB y permitir conexiones simultaneas. Estos procesos se detallan en el manual de SPICE.

Para configurar el uso del protocolo SPICE será necesario añadir los siguientes elementos al fichero XML:

- Añadir una nueva interfaz gráfica o modificar la existente y añadir un controlador de video de tipo QxL en elemento devices con la siguiente información:

```
  <devices>
  ..
   <graphics type='spice' port='5902' autoport='no' listen='0.0.0.0'
passwd='password' defaultMode='insecure'>
        <listen type='address' address='0.0.0.0'/>
   </graphics>
  <video>
      <model type='qxl'/>
  </video>
  ..
  </devices>
```

Los atributos usados para el elemento graphics son:

Atributo	Opcional	Explicación
type='spice'	No	Indica que se hará uso del protocolo SPICE para la interfaz gráfica.
port='5902'	No	Indica el puerto TCP que no hará uso de TLS en la comunicación remota. Si el valor fuese -1 indicará que se debe autoasignar al arrancar la máquina virtual.
autoport='no'	No	Indica si la será necesario autoasignará uno de los puertos TCP para la comunicación remota al arrancar la máquina virtual.
listen='0.0.0.0'	Si	Indica a la IP o nombre de host a la cual estará asociado los puertos de escucha TCP. De especificarse debe ser coherente con los atributos del elemento hijo listen. De no incluirse el valor del atributo type del elemento hijo listen deberá ser network, esto provoca la IP o nombre de host se calcule automáticamente en función de la subred seleccionada.
passwd='password'	Si	Indica la contraseña de autentificación de la conexión. Esta no está cifrada. Por defecto no se solicitará ninguna al establecer la conexión.
defaultMode='insecure'	Si	Indica que canal se usará para establecer la conexión. Los posibles valores son insecure (hace uso del canal sin cifrado por TLS), secure (hace uso del canal con cifrado por TLS) o any (que intentará establecer la conexión por el canal con cifrado por TLS y si no puede lo hará por el canal sin cifrado).

Los atributos usados para el elemento listen son:

Atributo	Opcional	Explicación
type='address'	No	Indica que como se establecerá la dirección de escucha. Lo posible valores son address (indica que especificaremos la dirección o nombre de hosts), network (indica que especificaremos la subred y a partir de ella se calculará la dirección o nombre de hosts), socket (indica que se asociará a un socket de Unix) o none (que indica que solo será accesible desde herramientas como virt-manager o virt-viewer de manera local)
address='0.0.0.0'	Si	Indica a la IP o nombre de host a la cual estará asociado los puertos de escucha TCP. Únicamente se hará uso cuando el valor de type sea address, en caso de ser otro se deberá especificar el atributo indicado.

EL atributo usado para el elemento model es:

Atributo	Opcional	Explicación
type='qxl'	No	Indica el tipo de controlador de video.

Si se quisiera hacer uso de conexiones cifrada mediante TLS será necesario modificar el fichero de configuración /etc/libvirt/qemu.conf con la siguiente información:

```
spice_tls = 1
spice_tls_x509_cert_dir = "certificado_x509"
```

Con lo que habilitaremos el uso de TLS y expecificaremos la ruta donde se aloja el certificado x509 que hará uso TLS. De no especificar la ruta el certificado deberá encontrarse en la ruta /etc/pki/libvirt-spice.

- Añadir los dispositivos USB que se quieran disponer para la conexión remota en elemento devices con la siguiente información:

```
<devices>
..
<controller type='usb' index='0' model='nec-xhci'/>
<redirdev bus='usb' type='spicevmc'/>
<redirdev bus='usb' type='spicevmc'/>
<redirdev bus='usb' type='spicevmc'/>
..
</devices>
```

Los atributos usados para el elemento controller son:

Atributo	Opcional	Explicación
type='usb'	No	Indica que se hará uso del protocolo SPICE para la interfaz gráfica.
index='0'	No	Indica el puerto TCP que no hará uso de TLS en la comunicación remota. Si el valor fuese -1 indicará que se debe autoasignar al arrancar la máquina virtual.
model='nec-xhci'	NO	Indica el tipo de controlador USB que se añadirá a la máquina virtual. El valor actual indica que será un controlador USB 3.0.

Los atributos usados para el elemento redirdev son:

Atributo	Opcional	Explicación
bus='usb'	No	Indica que tipo de dispositivo se va a redirigir.
type='spicevmc'	No	Indica el protocolo por el cual se va a redirigir el dispositivo.

Se deberá añadir un elemento redirdev por cada dispositivo USB que quiera conectar remotamente.

- Añadir la declaración de la variable de entorno para habilitar el uso de conexiones simultaneas, para ello es necesario la siguiente información en el elemento domain:

```
  <domain type='kvm'
xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
  ...
  ...
  <qemu:commandline>
      <qemu:env name='SPICE_DEBUG_ALLOW_MC' value='1'/>
    </qemu:commandline>
  </domain>
```

Los atributos usados para el elemento domain son:

Atributo	Opcional	Explicación
type='kvm'	No	Indica el hipervisor que se usará.
xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'	No	Declara un nuevo espacio de nombre que se usará para poder establecer variable de entorno para la ejecución de qemu.

Los atributos usados para el elemento qemu:env hijo del elemento qemu:commandline son:

Atributo	Opcional	Explicación
name='SPICE_DEBUG_ALLOW_MC'	No	Indica el nombre de la variable de entorno necesaria para habilitar el uso de conexiones simultaneas.
value='1'	No	Indica el valor que se le asignará a la variable de entorno.

La funcionalidad de conexiones simultaneas está en fase experimental y puede que provocar un funcionamiento incorrecto entre conexiones con diferente ancho de banda.

3.2.4 Líneas de continuación

Una posible línea es el estudio de la configuración y uso de otras características de SPICE como puede la compartición de carpetas en remoto o el uso de compartido del portapapeles.

3.3 Clúster de alta disponibilidad

En esta sección se detallará una solución de alto disponibilidad basada en el uso de las tecnologías pacemaker[50], corosync[51] y drbd[52].

Las tecnologías pacemaker y corosync serán las encargadas de controlar y administrar el clúster y sus recursos configurados. Entre alguna de las tereas que realizan es comprobar la conectividad entre los nodos, el estado del recurso levantado en un servidor concreto, mover o activar un recurso en otro servidor si falla en el servidor que se encuentra actualmente.

La tecnología drbd permitirá tener replicado y sincronizada la información entre varios nodos y mantener la funcionalidad de algunos de los recursos que se base en el uso de almacenamiento si fuera necesario cambiar el servidor en el que se está ejecutando. Ejemplos de recurso que necesitará el uso de drbd pueden ser un gestor de base de datos o un servidor de archivos como nfs.

Las tecnologías pacemaker y corosync será usada y configurada mediante la herramienta pcs[53], esta permite la administración mediante línea de comando o vía web.

3.3.1 Situación actual y problemática

En el laboratorio de Telemática se dispone dos máquinas virtuales las cuales están configurados para proporcional todos los servicios necesarios para el funcionamiento de este.

Estas se encuentran en una configuración en la que existe una maquina principal que se encuentra siempre activa quien proveer los servicios y otra secundaria que se encuentra pausada y que es una copia de la principal donde se adapta la configuración de red para no colisionar con la principal.

Con esta configuración si la maquina principal fallará o dejara de responder uno de sus servicios es necesario realizar la activación de la maquina secundaria manualmente. Además, existe la problemática de la maquina secundaria debe ser actualizada constantemente cada vez que se realizan cambios sobre uno de los servicios que se utilizan, pudiendo cambiar la configuración o el contenido ofrecido por alguno de los servicios.

Ilustración 3-39 Distribución actual del laboratorio

Se quiere adaptar una solución que permita recuperarse automáticamente ante la caída de una maquina completamente o de un servicio y de esta manera minimizar el impacto sobre la experiencia del usuario al hacer uso del laboratorio.

Para poder abordar esta problemática existen multitud de soluciones entre ellas se puede citar algunas tecnologías como puede ser la orquestación de contenedores mediante Docker Swarm o Kubernertes o plataformas de computación de la nube como Opennbula.

Estas fueron descartadas como solución actual debido a la gran cohesión que posee Opengnsys[54] con otros servicios para poder desarrollar completamente sus funcionalidades, esto implica que no se pueda ubicar en diferentes maquinas o nodos servicios como el servidor web o samba. Este servicio es crucial para el funcionamiento del laboratorio debido a que es el usado para gestionar y administrar los equipos del laboratorio.

3.3.2 Solución elegida

Para detallará la solución adoptada el escenario usado estará compuesto por dos nodos en los cuales se encuentra configurado el sistema operativo Debian 10 buster y cuyo nombre serán servidor1 y servidor2 y sus direcciones IP serán IP1 e IP2 respectivamente asignada a la interfaz con nombre eth0.

Se dispondrá de una dirección IP IPcompartida será una dirección que será balanceada entre los nodos.

En ambos nodos se tiene una distribución de disco en el cual se tiene un disco con las siguientes particiones:

- /dev/sda1 donde se aloja la raíz del sistema operativo.
- /dev/sda3 que se usará para alojar las imágenes de Opengnsys.
- /dev/sda2 que una partición extendida donde se ubican las particiones /dev/sda5, /dev/sda6 y /dev/sda7.
- /dev/sda5 es la partición swap del sistema.
- /dev/sda6 que se usará para alojar los datos para la base datos.
- /dev/sda7 que se usará para alojar los datos para el servidor TFTP.

Todo el proceso se realizará con permiso de superusuario, en ambos nodos se encuentra instalado y configurado Opengnsys y el firewall usado está basado el uso de iptables[55].

En la solución adoptada se centran en la configuración del clúster para el correcto funcionamiento de Opengnsys.

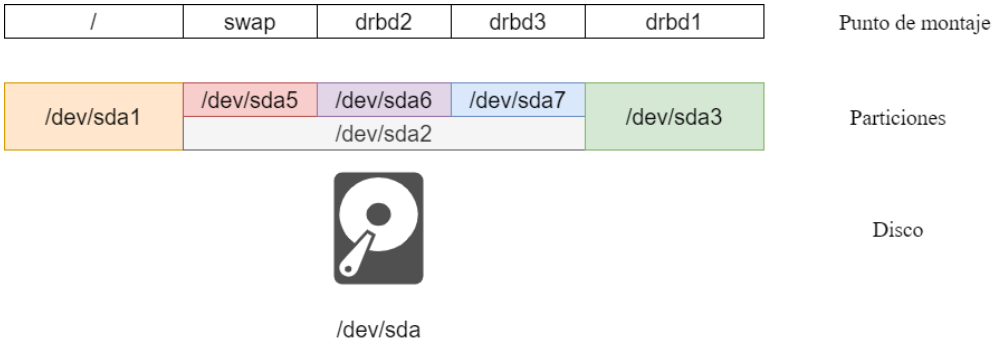

Ilustración 3-40 Distribución de almacenamiento en cada nodo del clúster.

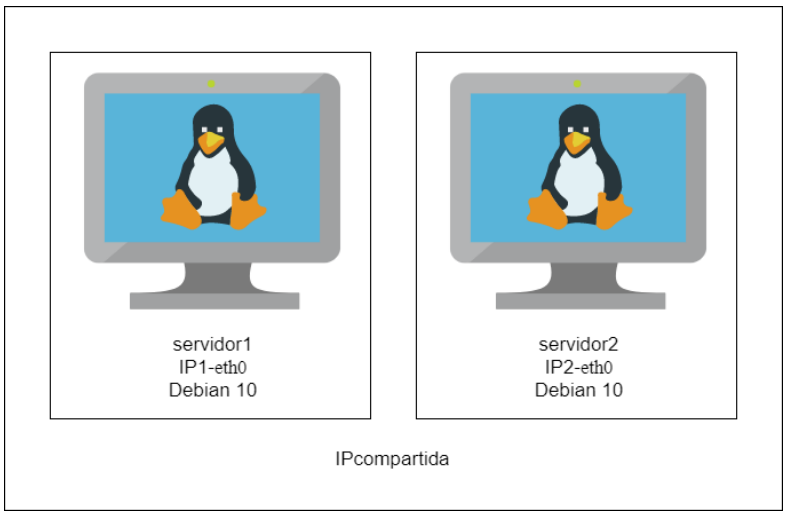

Ilustración 3-41 Distribución del clúster en la solución

La solución adoptada se divide en cuatros procesos distintos:

- Instalación y configuración inicial del clúster de alta disponibilidad.
- Instalación y configuración del almacenamiento distribuido y replicado.
- Adaptación de Opengnsys para su uso en el clúster de alta disponibilidad.
- Configuración de los servicios del clúster.

3.3.2.1 Instalación y configuración inicial del clúster de alta disponibilidad.

Se va a detallar la configuración inicial que se debe realizar para crear un clúster basada en el uso de pacemaker y corosync y administrado mediante línea de comando mediante el uso de la herramienta pcs.

Los pasos por seguir son:

1. Instalar las dependencias necesarias mediante el uso de la herramienta apt[56].

```
servidor1#apt install -y pacemaker pcs psmisc policycoreutils-
python-utils
servidor2#apt install -y pacemaker pcs psmisc policycoreutils-
python-utils
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
pacemaker pcs psmisc policycoreutils-python-utils	Nombre los paquetes de las dependencias necesarias.

2. Permitir el acceso a los puertos necesarios para pcs, pacemaker y corosync. Para ellos se creará nuevas reglas iptables.

```
servidor1#iptables -A INPUT -p tcp –match multiport --dport
2224,3121,5403,9929,21064 -j ACCEPT
servidor1#iptables -A INPUT -p udp --match multiport --dport
5404,5405,9929 -j ACCEPT
servidor2#iptables -A INPUT -p tcp –match multiport --dport
2224,3121,5403,9929,21064 -j ACCEPT
servidor2#iptables -A INPUT -p udp --match multiport --dport
5404,5405,9929 -j ACCEPT
```

Los argumentos usados serán:

-A	Añade la regla al final de la cena que se especifique a continuación.
INPUT	Indica la cadena sobre la que se establecerá la regla. Al solo indicar la cadena se establecerá la regla en la cadena de filtrado por la cual pasan los paquetes antes de ser entregado a un proceso local.
-p X	Establece sobre que protocolo se aplicara la regla. El valor de X será udp o tcp.
–match multiport	Indica que se va a establecer más de un puerto para la regla.
--dport Y	Establece los puertos de destino sobre los cuales le aplicara la acción seleccionada. El valor de Y será aquello puerto que se desea aceptar para el protocolo indicado con el valor de X.
-j ACCEPT	Indica la acción la cual se aplicará para esta regla será la de aceptar el paquete.

Esto puerto son especificado en la página oficial de Red Hat sobre la configuración de iptables para el clúster de alta disponibilidad[57]

3. Habilitar para que se autoarranque el servicio de pcsd al iniciar. Para ello se usará de la herramienta systemctl[58].

```
servidor1#systemctl enable pcsd.service
servidor2#systemctl enable pcsd.service
```

Los argumentos usados serán:

enable	Establece que el servicio se autoarranque al iniciar el sistema.
pcsd.service	Indica el nombre del servicio que se desea configurar.

4. Iniciar el servicio de pcsd. Para ello se usará de la herramienta systemctl.

```
servidor1#systemctl start pcsd.service
servidor2#systemctl start pcsd.service
```

Los argumentos usados serán:

start	Indica que se inicie un servicio.
pcsd.service	Indica el nombre del servicio que se desea iniciar.

5. Asignar la contraseña al usuario hacluster que será el usuario cuando para establecer la comunicación del clúster, esta contraseña debe coincidir en los nodos. Para ello haremos uso de la herramienta passwd[59], con la cual será necesario introducir dos veces la contraseña elegida.

```
servidor1#passwd hacluster
servidor2#passwd hacluster
```

Los argumentos usados serán:

hacluster	Indica el nombre de usuario que se desea establecer la contraseña.

6. Modificar el fichero de hosts(/etc/hosts) para añadir en ambos nodos la traducción del nombre de ambos nodos con la dirección IP con la que se comunicaran dentro del clúster. Se añadirá al fichero las siguientes líneas.

```
servidor1 IP1
servidor2 IP2
```

7. Autentificación de los demonios pcsd locales contra los otros demanios de los nodos en el clúster para ello se usará la herramienta pcs. Se deberá introducir la contraseña establecida para el usuario hacluster.

```
servidor1#pcs host auth servidor1 servidor2
```

Los argumentos usados serán:

host auth	Indica que se va a establecer la autentificación entre los nodos que formaran el clúster.
servidor1 servidor2	Nombre de los nodos que se autentificaran entre sus demonios pcsd.

8. Crear clúster con los nodos que lo formará para ello se usará la herramienta pcs.

```
servidor1#pcs cluster setup clusterHA servidor1 servidor2
```

Los argumentos usados serán:

cluster setup	Indica que se va a crear un nuevo clúster.
clusterHA	Indica el nombre que se establecerá al clúster.
servidor1 servidor 2	Nombre de los nodos que formará el clúster.

9. Iniciar todos los nodos del clúster para ello se usará la herramienta pcs.

```
servidor1#pcs cluster start servidor1 servidor2
```

Los argumentos usados serán:

cluster start	Indica que se va a iniciar los nodos en el clúster.
servidor1 servidor2	Nombre de los nodos que se iniciarán.

10. Habilitar para que se autoarranque los servicios corosync y pacemaker al iniciar, estos serán usado por pcsd para el funcionamiento del clúster. Para ello se usará de la herramienta systemctl.

```
servidor1#systemctl enable corosync.service
servidor1#systemctl enable pacemaker.service
servidor2#systemctl enable corosync.service
servidor2#systemctl enable pacemaker.service
```

Los argumentos usados serán:

enable	Establece que el servicio se autoarranque al iniciar el sistema.
corosync.service o pacemaker.service	Indica el nombre del servicio que se desea configurar.

11. Desactivar el servicio la política STONIT ya que para el clúster no será necesario debido a que no se necesitan acceder simultáneamente a un conjunto de ficheros desde los dos nodos. Para ello se usará de la herramienta pcs.

```
servidor1#pcs property set stonith-enabled=false
```

Los argumentos usados serán:

property set	Indica que se va establece un valor de una propiedad del clúster.
stonith-enabled=false	Indica que no se hará uso de la política de STONIT.

12. Configurar como actuará el clúster en caso de no haber quorum entre los nodos, para un servidor con solo dos nodos ocurrirá cuando uno de ellos no esté accesible. Debido a esto se establece que se ignore y se permita levantar y administrar los recursos configurado. Para ello se usará de la herramienta pcs.

```
servidor1#pcs property set no-quorum-policy=ignore
```

Los argumentos usados serán:

property set	Indica que se va establece un valor de una propiedad del clúster.
no-quorum-policy=ignore	Indica que no se ignora el hecho de que no exista quorum entre los nodos.

13. Configurar para que el clúster verifique si hubo cambio cada 30 segundos. Para ello se usará de la herramienta pcs.

```
servidor1#pcs property set cluster-recheck-interval=30s
```

Los argumentos usados serán:

property set	Indica que se va establece un valor de una propiedad del clúster.
cluster-recheck-interval=30s	Indica que se haga una verificación cada 30 segundos.

14. Comprobar el estado del clúster mediante la herramienta pcs.

```
servidor1#pcs status
```

Los argumentos usados serán:

status	Indica que se muestre el estado clúster.

15. Configurar el agente de recurso heartbeat de tipo anything[60] que permite la administración de un servicio personalizado en el clúster, un ejemplo puede ser el lanzado de un script que nos permita configurar un elemento concreto del sistema de un nodo.
    a. Descargar el agente de mediante la herramienta curl[61].

```
curl https://raw.githubusercontent.com/ClusterLabs/resource-
agents/master/heartbeat/anything >
/usr/lib/ocf/resource.d/heartbeat/anything
```

Los argumentos usados serán:

https://raw.githubusercontent.com /ClusterLabs/resource-agents/master/heartbeat/anything	Indica URL desde donde se descargará el agente de recurso anything.
/usr/lib/ocf/resource.d/heartbeat/anything	Indica la ubicación donde de almacenera el agente de recurso anything.

    b. Cambiar los permisos adecuada al agente de recurso mediante la herramienta chmod.

```
chmod 0755 /usr/lib/ocf/resource.d/heartbeat/anything
```

Los argumentos usados serán:

0755	Adjudica permiso de lectura y ejecución para cualquier usuario y solo de escritura para el usuario de superusuario.
/usr/lib/ocf/resource.d/heartbeat/anything	Indica la ubicación donde del agente de recurso anything.

3.3.2.2 Instalación y configuración del almacenamiento distribuido y replicado

A continuación, se detallará como crear y configurar las particiones basada en la tecnología drbd que será usada para contener los datos que será compartido entre los nodos del clúster para el funcionamiento de Opengnsys.

Mediante el uso de tecnología drbd se dispondrá de dispositivos de bloques que se sincronizará de forma automática en diferentes nodos. Esta tecnología está integrada en el kernel de Linux.

Una de las principales desventajas de esta tecnología el consumo de red cuando es necesario sincronizar grandes cantidades de datos.

Los pasos por seguir son:

1. Instalar las dependencias necesarias mediante el uso de la herramienta apt.

```
servidor1#apt install drbd-utils
servidor2#apt install drbd-utils
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
drbd-utils	Nombre el paquete de la dependencia necesaria.

2. Cargar el módulo drbd al kernel Linux mediante el uso de la herramienta modprobe[62].

```
servidor1#modprobe drbd
servidor2#modprobe drbd
```

El argumento usado será:

drbd	Indica el módulo que se desea activar.

3. Configurar para que se cargue automáticamente el módulo de drbd al kernel de Linux al iniciar el sistema operativo. Para ello se creará un fichero que indicará que modulo se debe a cargar al iniciar al introducirlo en el directorio /etc/modules-load.d/.

```
servidor1#echo drbd > /etc/modules-load.d/drbd.conf
servidor2#echo drbd > /etc/modules-load.d/drbd.conf
```

Los argumentos usados serán:

drbd	Indica el módulo que se desea activar.
/etc/modules-load.d/drbd.conf	Crea un fichero que indica el módulo a cargar al iniciar el sistema.

4. Crear fichero que define los diferentes dispositivos de bloques que se va a crear en los nodos. Para ello se va a crear en ambos nodos el fichero /etc/drbd.d/recursos.res con el siguiente contenido.

```
resource imagenes {
    meta-disk internal;
    device /dev/drbd1;
    disk /dev/sda3;
    syncer {
     rate 80M;
    }
    net {
     allow-two-primaries;
     after-sb-0pri discard-zero-changes;
     after-sb-1pri discard-secondary;
     after-sb-2pri disconnect;
    }
    on servidor1 { address IP1:7789; }
    on servidor2 { address IP2:7789; }
}

resource basedatos {
    meta-disk internal;
    device /dev/drbd2;
    disk /dev/sda6;
    syncer {
     rate 80M;
    }
    net {
     allow-two-primaries;
     after-sb-0pri discard-zero-changes;
     after-sb-1pri discard-secondary;
     after-sb-2pri disconnect;
   }
    on servidor1 { address IP1:7790; }
    on servidor2 { address IP2:7790; }
}

resource tftp {
    meta-disk internal;
    device /dev/drbd3;
    disk /dev/sda7;
    syncer {
     rate 80M;
    }
    net {
     allow-two-primaries;
     after-sb-0pri discard-zero-changes;
     after-sb-1pri discard-secondary;
     after-sb-2pri disconnect;
    }
    on servidor1 { address IP1:7791; }
    on servidor2 { address IP2:7791; }
}
```

El contenido de este fichero será detallado en el ANEXO G: Clúster de alta Disponibilidad

5. Permitir el acceso a los puertos necesarios para los diferentes recursos que se van a crear. Para ellos se creará nuevas reglas iptables.

```
 servidor1#iptables -A INPUT -p tcp --match multiport --dport
7789,7790,7791 -j ACCEPT
 servidor2#iptables -A INPUT -p tcp --match multiport --dport
7789,7790,7791 -j ACCEPT
```

Los argumentos usados serán:

-A	Añade la regla al final de la cena que se especifique a continuación.
INPUT	Indica la cadena sobre la que se establecerá la regla. Al solo indicar la cadena se establecerá la regla en la cadena de filtrado por la cual pasan los paquetes antes de ser entregado a un proceso local.
-p X	Establece sobre que protocolo se aplicara la regla. El valor de X será udp o tcp.
–match multiport	Indica que se va a establecer más de un puerto para la regla.
--dport Y	Establece los puertos de destino sobre los cuales le aplicara la acción seleccionada. El valor de Y será aquello puerto que se desea aceptar para el protocolo indicado con el valor de X.
-j ACCEPT	Indica la acción la cual se aplicará para esta regla será la de aceptar el paquete.

6. Eliminar el formato de las particiones para posteriormente aplicar el formato drbd posteriormente, para realizarlo se sobrescribe los primeros 1Mb. Para ello se hará uso de la herramienta dd.

```
servidor1#dd if=/dev/zero of=/dev/sda3 bs=1M count=1
servidor1#dd if=/dev/zero of=/dev/sda6 bs=1M count=1
servidor1#dd if=/dev/zero of=/dev/sda7 bs=1M count=1
servidor2#dd if=/dev/zero of=/dev/sda3 bs=1M count=1
servidor2#dd if=/dev/zero of=/dev/sda6 bs=1M count=1
servidor2#dd if=/dev/zero of=/dev/sda7 bs=1M count=1
```

Los argumentos usados serán:

if=/dev/zero	Selecciona como dispositivo origen un dispositivo virtual que genera únicamente 0.
of=/dev/sda3	Selecciona como dispositivo el dispositivo que al que se desea eliminar el formato.
bs=1M	Indica que leerá y escribirá bloques de cuyo tamaño es de 1Mb.
count=1	Indica que solo se va a leer y escribir un solo bloque de datos.

7. Inicializar los metadatos en las diferentes particiones asociados a los diferentes dispositivos de bloques configurados. Para ello se hará uso de la herramienta drbdadm[63].

```
servidor1#drbdadm create-md imagenes
servidor1#drbdadm create-md basedatos
servidor1#drbdadm create-md tftp
servidor2#drbdadm create-md images
servidor2#drbdadm create-md basedatos
servidor2#drbdadm create-md tftp
```

Los argumentos usados serán:

create-md	Indica que se inicialice los metadatos en el dispositivo asociado al dispositivo de bloque indicado.
images, basedatos o tftp	Indica el dispositivo de bloque.

8. Levantar los diferentes dispositivos de bloques configurado, al realizar las operaciones necesario para iniciar la sincronización entre los diferentes nodos. Para ello se hará uso de la herramienta drbdadm.

```
servidor1#drbdadm up imagenes
servidor1#drbdadm up basedatos
servidor1#drbdadm up tftp
servidor2#drbdadm up images
servidor2#drbdadm up basedatos
servidor2#drbdadm up tftp
```

Los argumentos usados serán:

up	Indica que se comience la sincronización entre los diferentes nodos para el dispositivo indicado
images, basedatos o tftp	Indica el dispositivo de bloque.

9. Establecer la información de que nodo será usada para sobrescribir en el resto de los nodos. Para ello se hará uso de la herramienta drbdadm.

```
servidor1#drbdadm -- --overwrite-data-of-peer primary imagenes
servidor1#drbdadm -- --overwrite-data-of-peer primary basedatos
servidor1#drbdadm -- --overwrite-data-of-peer primary tftp
```

Los argumentos usados serán:

--	Indica que se va a indicar una opción de configuración de recurso
--overwrite-data-of-peer	Establece que se sobrescriba la información del nodo actual sobre el resto de los nodos.
primary	Establece el nodo actual como un nodo primario.
images, basedatos o tftp	Indica el dispositivo de bloque.

10. Establecer el otro nodo como nodo primario también. Para ello se hará uso de la herramienta drbdadm.

```
servidor2#drbdadm primary imagenes
servidor2#drbdadm primary basedatos
servidor2#drbdadm primary tftp
```

Los argumentos usados serán:

primary	Establece el nodo actual como un nodo primario.
images, basedatos o tftp	Indica el dispositivo de bloque.

11. Comprobar el estado de actual de sincronización de los diferentes dispositivos de bloques. Será necesario que ambos nodos se encuentren en estado UpDate ya que indicará que se encuentra sincronizado. Para ello consultaremos el estado mediante la información almacenada en el fichero especial /proc/drbd.

```
servidor1#cat /proc/drbd
```

El resultado debería se parecido al siguiente resultado.

```
root@servidor1:~# cat /proc/drbd
version: 8.4.10 (api:1/proto:86-101)
srcversion: 473968AD625BA317874A57E

 1: cs:Connected ro:Primary/Primary ds:UpToDate/UpToDate C r-----
    ns:8208 nr:116 dw:152 dr:854693 al:2 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
 2: cs:Connected ro:Primary/Primary ds:UpToDate/UpToDate C r-----
    ns:14620 nr:39960 dw:53624 dr:8365 al:6 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
 3: cs:Connected ro:Primary/Primary ds:UpToDate/UpToDate C r-----
    ns:4 nr:4100 dw:4104 dr:213 al:1 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

Ilustración 3-42 Resultado comprobación estado sincronización drbd

En este resultado podemos observar que todos los dispositivos administrados con drbd se encuentra sincronizado y los nodos se encuentra conectados.

Se deberá espera mientras que el resultado del anterior comando dando como resultado uno de los estados sea Inconsistent, esto puede demorarse bastante tiempo dependiendo de la velocidad de la red y de la cantidad de datos a sincronizar.

12. Formatear los dispositivos de bloques generados con el sistema de fichero ext3, el sistema de fichero es totalmente opcional. Para ello haremos uso de la herramienta mkfs[64].

```
servidor1#mkfs.ext3 /dev/drbd1
servidor1#mkfs.ext3 /dev/drbd2
servidor1#mkfs.ext3 /dev/drbd3
```

El argumento usado será:

/dev/drbdX	Indica el dispositivo de bloque al cual se formateará con el sistema de fichero ext3.

13. Habilitar para que se autoarranque el servicio de drbd al iniciar. Para ello se usará de la herramienta systemctl.

```
servidor1#systemctl enable drbd.service
servidor2#systemctl enable drbd.service
```

Los argumentos usados serán:

enable	Establece que el servicio se autoarranque al iniciar el sistema.
drbd.service	Indica el nombre del servicio que se desea configurar.

3.3.2.3 Adaptación de Opengnsys para su uso en el clúster de alta disponibilidad

Para el correcto funcionamiento de opengnsys ser debe realizar una serie de modificaciones y procedimientos que se comentará sobre la versión 1.1.1c de opengnsys.

El primer procedimiento será modificar una serie de fichero en ambos nodos:

- Permitir el acceso de la base de datos desde una dirección IP externa al nodo. Para ello será necesario ejecutar la siguiente sentencia en la base de datos.

```
GRANT ALL ON ogAdmBD.* to 'usuarioOgAdmBD'@'IPexterna' IDENTIFIED
BY 'contrasenaUsuarioOgAdmBD' WITH GRANT OPTION;
flush privileges;
```

El valor usuarioOgAdmBD deberá ser sustituido por el usuario configurado en la instalación de Opengnsys como usuario de acceso a base de datos, también se puede consultar en fichero de configuración /opt/opengnsys/etc/ogAdmServer.cfg en la variable USUARIO.

El valor IPexterna deberá ser sustituido por las diferentes direcciones IP que deben poder tener acceso a la base de datos, para nuestro escenario seri IP1, IP2 e IPcompartida.

- Modificar el fichero del servicio de opengnsys para añadir la posibilidad de ver el estado del servicio mediante el argumento status. Para ello se modificar el fichero /etc/init.d/opengnsys. Se añade el siguiente fragmento tras la función llamada para_demonios, con el cual implementa un método para poder comprobar los diferentes servicios o aplicaciones levantados al iniciar el servicio de opengnsys.

```
estado_demonios() {
  STATUSSTOP="FALSE"
  # Compure los servicios indicados.
  if [ $RUN_OGADMSERVER = "yes" ]; then
     $ACTIONMSG "Comprobando demonio: $SERVERNAME"
     $DEAMONSTATUS $SERVERNAME
     if [ $? = 0 ]; then
         $SUCCESSMSG
     else
        $FAILMSG
        STATUSSTOP="TRUE"
     fi
     $ACTIONMSG "Comprobando demonio: $SERVERAUXNAME"
     auxstatus "$SERVERAUXNAME"    #   NUEVO
      if [ $? = 0 ]; then
         $SUCCESSMSG
     else
```

```
        $FAILMSG
        STATUSSTOP="TRUE"
    fi
fi
if [ $RUN_OGADMREPO = "yes" ]; then
    $ACTIONMSG "Comprobando demonio: $REPOAUXNAME"
    auxstatus "$REPOAUXNAME"
     if [ $? = 0 ]; then
        $SUCCESSMSG
    else
        $FAILMSG
        STATUSSTOP="TRUE"
    fi
fi

if [ $RUN_OGADMAGENT = "yes" ]; then
    $ACTIONMSG "Comprobando demonio: $AGENTNAME"
    $DEAMONSTATUS $AGENTNAME
     if [ $? = 0 ]; then
        $SUCCESSMSG
    else
        $FAILMSG
        STATUSSTOP="TRUE"
    fi
fi

if [ $RUN_BTTRACKER = "yes" ]; then
    $ACTIONMSG "Comprobando demonio: $BTTRACK"
    if [ -e $BTTRACKPID ]; then
     $TRACKERSTATUS
     if [ $? = 0 ]; then
       $SUCCESSMSG
     else
       $FAILMSG
       STATUSSTOP="TRUE"
     fi
  else
       $FAILMSG
       STATUSSTOP="TRUE"
    fi
fi

if [ $RUN_BTSEEDER = "yes" ]; then
  $ACTIONMSG "Comprobando demonio: $BTSEEDER"
  if [ -e $BTSEEDERPID ]; then
    $SEEDERSTATUS
    if [ $? = 0 ]; then
       $SUCCESSMSG
   else
       $FAILMSG
       STATUSSTOP="TRUE"
    fi
  else
    $FAILMSG
       STATUSSTOP="TRUE"
  fi
```

```
  fi

  if [ STATUSSTOP = "TRUE" ]; then
    return 3
  else
    return 0
  fi
}

auxstatus() {
  return $(ps x | grep -v 'grep'| grep  -E 'faucet' | grep -q $1)
}

auxstatus() {
  return $(ps x | grep -v 'grep'| grep  -E 'faucet' | grep -q $1)
}

config

case "$1" in
  start)
        arranca_demonios
        ;;
  stop)
        para_demonios
        ;;
  status)
        estado_demonios
        exit $?
        ;;
  restart)
        para_demonios
        arranca_demonios
        ;;

  *)
        echo "Uso: $0 {start|stop|restart|status}"
        exit 1
        ;;
esac

exit 0
```

- Modificar el fichero de funciones comunes usados en script del servidor y repositorio de opengnsys para contemplar que la base datos se encuentre otro host. La dirección de la base de datos se obtendrá de la variable datasource del fichero de configuración /opt/opengnsys/etc/ogAdmServer.cfg.

  Se sustituirá la función dbexec el fichero /opt/opengnsys/lib/ogfunctions.sh con el siguiente contenido.

```
function dbexec () {
    MYCNF=$(mktemp)
    trap "rm -f $MYCNF" 0 1 2 3 6 9 15
    touch $MYCNF
    chmod 600 $MYCNF
    cat << EOT > $MYCNF
```

```
  [client]
  user=$USUARIO
  password=$PASSWORD
  host=$datasource
  EOT
      mysql --defaults-extra-file="$MYCNF" -D "$CATALOG" -s -N -e "$1"
|| \
          raiseError access "Cannot access the databse"
      rm -f "$MYCNF"
  }
```

- Modificar el script que levanta el tracker de Torrent periódicamente para periódicamente del proceso de cron que es configurado al realizar la instalación de Opengnsys para que el valor de la variable que almacena el PID se actualice correctamente. Para ello se añadir el siguiente fragmento al final del fichero /opt/opengnsys/bin/torrent-tracker.

```
  pgrep bttrack > $BTTRACKPID
```

- Modificar el script que cambia la dirección IP por defecto de Opengnsys para contemplar que una interfaz posea más de una dirección IP asociada. Para ello se modifica el bucle que detecta la interfaz red y/o la dirección IP para realizar las modificaciones en Opengnsys sobre el script /opt/opengnsys/bin/setserveraddr. Se debe sustituir el fragmento que se encuentra entre los comentarios los comentarios con el siguiente fragmento:
    - # Detecting network interfaces.
    - # Checking if IP address has been detected.

```
  DEVICES=$(ip -o link show up | awk -F: '$2!~/lo/ {print $2}')
  for DEV in $DEVICES; do
           # If the network interface is found, get its IP address.
   IP=$(ip -o addr show dev "$DEV" | awk '$3~/inet$/ {sub (/\/.*/, "");
print ($4)}')
      if echo "$IP" | grep -q "$1"; then
       SERVERDEV="$DEV"
       SERVERIP="$1"
      fi
      if [ "$DEV" == "$1" ]; then
          SERVERDEV="$DEV"
       if [ "$(echo "$IP"| wc -w)" != "1" ]; then
          while  ! echo "$IP" | grep -q "^${SERVERIP}$"; do
            echo  "La  interfaz  elgida  tiene  varias  direcciones  IP
asociadad debes elegir una "
            echo "$IP"
            echo -ne "Introduce la direccion IP: "
            read -r SERVERIP
          done
      else
         SERVERIP="$IP"
      fi
     fi
  done
```

Este fragmento solicitará seleccionar una dirección IP si se detecta que la interfaz introducida como argumento

posee más de una dirección IP asociada. Si por contrario el argumento fuera una dirección IP comprueba que esta pertenece a alguna de las interfaces a pesar de que no sea la única.

- Sincronizar el valor de los tokens de los ficheros de configuración para que el valor sea el mismo en ambos nodos. Los ficheros los cuales se deberán modificar para que sean iguales son:
    - /opt/opengnsys/etc/ogAdmRepo*.cfg: Donde el valor de la variable ApiToken del nodo servidor1 se deberá sobrescribir en el nodo servidor2.
    - /opt/opengnsys/etc/ogAdmServer*.cfg: Donde el valor de la variable APITOKEN del nodo servidor1 se deberá sobrescribir en el nodo servidor2

En el segundo procedimiento consistirá en cambiar la dirección IP por defecto que usa Opengnsys y sincronizar la configuración de este entre los nodos del clúster. Los pasos por seguir son:

1. Añadir la dirección IP IPcompartida al interfaz eth0 para ello se hará uso de la herramienta ip[65].

```
servidor1#ip addr add IPcompartida/32 dev eth0
servidor2#ip addr add IPcompartida/32 dev eth0
```

Los argumentos usados serán:

addr add	Indica que se procederá a añadir una nueva dirección IP.
IPcompartida/32	Indica la dirección IP va a ser añadida.
dev eth0	Indica la interfaz de red a la cual se va a asociar la nueva dirección IP.

2. Lanzar el script /opt/opengnsys/bin/setserveraddr para realizar el cambio de dirección IP por defecto usada por Opengnsys.

```
servidor1#/opt/opengnsys/bin/setserveraddr IPcompartida
servidor2#/opt/opengnsys/bin/setserveraddr IPcompartida
```

El argumento usado será:

IPcompartida	Indica la IP que pasará a la dirección IP por defecto de Opengnsys.

3. Eliminamos la dirección IP IPcompartida para ello se hará uso de la herramienta ip.

```
servidor1#ip addr del IPcompartida/32 dev eth0
servidor2#ip addr del IPcompartida/32 dev eth0
```

Los argumentos usados serán:

addr del	Indica que se procederá a eliminar la dirección IP.
IPcompartida/32	Indica la dirección IP va a eliminada.
dev eth0	Indica la interfaz de red a la cual se va a eliminar la dirección IP.

En el tercer procedimiento consistirá en transferir los datos necesarios a los dispositivos de bloques creado con drbd. Los pasos por seguir son:

1. Crear los directorios temporales que se usará para montar los diferentes dispositivos y para la creación se usará la herramienta mkdir.

```
servidor1#mkdir /tmp/imagenes
servidor1#mkdir /tmp/basedatos
servidor1#mkdir /tmp/tftp
```

El argumento usado será:

/tmp/X	Indica el nombre del directorio.

2. Montar los dispositivos de bloques en las carpetas creadas anteriormente mediante el uso de la herramienta mount.

```
servidor1#mount /dev/drb1 /tmp/imagenes
servidor1#mount /dev/drb2 /tmp/basedatos
servidor1#mount /dev/drb3 /tmp/tftp
```

Los argumentos usados serán:

/dev/drbX	Indica el dispositivo a montar.
/tmp/Y	Indica el nombre del directorio donde se montará el dispositivo.

3. Copiar el contenido de los directorios donde posteriormente será montado los dispositivos de bloques para ello se utilizarla la herramienta rsync.

```
servidor1#rsync -Pa /opt/opengnsys/images/ /tmp/imagenes
servidor1#rsync -Pa /var/lib/mysql/ /tmp/bbdd/
servidor1#rsync -Pa /srv/tftp/ /tmp/tftp
```

Los argumentos usados serán:

-a	Indica que se quiere que se haga la copia recursivamente de los archivos que se encuentre rutaAbsoluta preservando las propiedades, fecha de modificación, propietarios y grupos de estos. Además, también se preservará lo enlaces simbólicos y recreará fichero de dispositivos y especiales como sockets o fifos. Es equivalente a usar los argumentos -rlptgoD.
-P	Indica que se realice la copia de manera que se puede ver el progreso de esta y además reanudar la copia en caso de que se interrumpiera. Es equivalente a usar los argumentos --partial --progress.
/opt/opengnsys/images/ /var/lib/mysql/ /srv/tftp/	Indica el nombre del directorio origen.
/tmp/Y	Indica el nombre del directorio destino.

4. Eliminar contenido de los directorios donde será montado los dispositivos de bloques para ello se utilizarla la herramienta rm.

```
servidor1#rm -R /opt/opengnsys/images/*
servidor1#rm -R /var/lib/mysql/*
servidor1#rm -R /srv/tftp/*
servidor2#rm -R /opt/opengnsys/images/*
servidor2#rm -R /var/lib/mysql/*
servidor2#rm -R /srv/tftp/*
```

Los argumentos usados serán:

-R	Indica que se haga de manera recursiva.
/opt/opengnsys/images/* /var/lib/mysql/* /srv/tftp/*	Indica el nombre del directorio que cuyo contenido será eliminado.

5. Desmontar los dispositivos de bloques mediante el uso de la herramienta umount[66].

```
servidor1#umount /tmp/imagenes
servidor1#umount /tmp/basedatos
servidor1#umount /tmp/tftp
```

El argumento usado será:

/tmp/Y	Indica el nombre del directorio donde se encuentra montado el dispositivo.

3.3.2.4 Configuración de los servicios del cluster

administrado por el clúster para el escenario propuesto.

En un principio se explicará algunos detalles necesarios para comprender la motivación de la solución adoptada en cuanto a configuración y distribución de los recursos.

Para realizar el montaje y la administración de los dispositivos de bloques creado con drbd se creará un recurso que se encargara que un solo nodo este como primario o maestro, de esta manera todas las operaciones sobre el nodo primario o maestro se sincronizasen correctamente en el secundario o esclavo. En conjunto se creará otro recurso que será el encargado de montar el bloque en el sistema de ficheros y esto se vinculará para que únicamente se ejecute en el nodo que actúe de primario o maestro, ya que es no es posible montar un el dispositivo si el nodo actúa como primario o maestro y nunca en dos nodos simultáneamente.

- Para aquello recursos que hacen uso de alguno de los dispositivos de bloques creado con drbd se ha creado un grupo para que tanto el recurso como el montaje del dispositivo siempre se realice sobre el mismo nodo.
- Debido a que algún recurso se puede encontrar en un nodo distinto al nodo donde se encuentra la IP compartida se hará uso del agente de recurso anything junto al script scriptReglaIptables.sh, que se detallará en elANEXO G: Clúster de alta Disponibilidad, con los cuales se redireccionará entre nodo si se recibe una petición cuyo destino es algún recurso determinado. El recurso creara las reglas de iptables para la redirección se encontrará siempre en el nodo donde no se esté ejecutando el recurso en cuestión.

- Se dispondrá de un grupo de recursos que estará compuesto por opengnsys y otros recursos que deben ejecutarse en el mismo nodo. Entre algunas de las dependencias detectada en las pruebas están:
    - Dirección IP compartida y servidor Apache: La dependencia es provocado debido a que para responder la petición del menú asociado a al cliente cuando inician ogLive de Opengnsys se va en la dirección IP origen del paquete que es procesado por Apache. Esto provoca que si se redirige esta petición no se le responderá al cliente con el menú asignado.
    - Dirección IP compartida y Opengnsys: La principal dependencia detectada se encuentra en cómo se desarrolla el proceso de restauración de una imagen mediante Torrent, al iniciar el proceso de descarga el cliente solicitar él envió de un determinado fichero mediante http al puerto 6969 y el servidor establece la comunicación contra un puerto que el cliente abre entre 2706 y el 2106. Esto provoca que si se redirige esta petición el servidor intentara establece la conexión contra uno de los puertos del nodo que redirigió el paquete y este no sabrá a que cliente debe reenviarlo o si debe reenviarlo.
    - Opengnsys, Samba y Rsync: La dependencia detectada se debe al proceso de creación de imágenes usado en Opengnsys debido a que el cliente envía operaciones al servicio de opengnsys como crear una imagen, montarla o reducirla que provoca la ejecución de acciones en el nodo que lo ejecuta mientras que la formación para crear la imagen es enviada mediante Samba o Rsync, dependiendo del tipo de imagen. Esto provoca que si no se encuentra todos los recursos en el mismo nodo se entraría enviando información a un nodo mientras que se realizan operaciones en otro nodo.

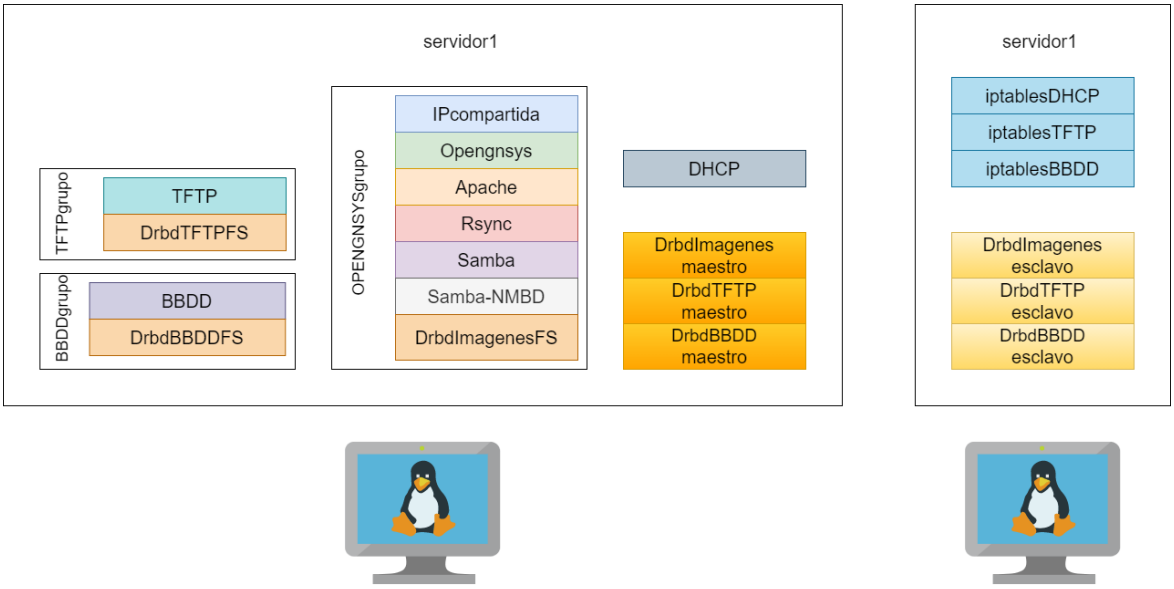

Ilustración 3-43 Distribución inicial de los recursos en el clúster.

Los pasos que se deben seguir son los siguientes:

1. Instalar las dependencias necesarias mediante el uso de la herramienta apt.

```
servidor1#apt -y install arping libxml2-utils
servidor2#apt -y install arping libxml2-utils
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
arping libxml2-utils	Nombre los paquetes de las dependencias necesarias.

2. Crear el recurso que permitirá uno de los nodos tener asignada la dirección IP IPcompartida en función de donde se ubique este mismo. Se hará uso de la herramienta pcs y el agente de recurso ocf:heartbeat:IPaddr2[67].

```
servidor1#pcs resource create IPcompartida ocf:heartbeat:IPaddr2
ip="IPcompartida" cidr_netmask="24" \
nic="eth0" flush_routes="true" arp_sender="iputils_arping" \
op monitor interval="10s"
```

Los argumentos usados serán:

resource create	Indica que se va a crear un nuevo recurso en clúster.
IPcompartida	Nombre que se le asignara al recurso.
ocf:heartbeat:IPaddr2	Indica que el agente con el que se hará de uso para el recurso. Este permite administrar direcciones IPv4 virtuales.
ip="IPcompartida	Establece la dirección IP virtual la cual se asignará a la interfaz de red cuando se ejecute el recurso en el nodo.
cidr_netmask="24"	Establece la mascará de subred que se usará junto a la dirección IP virtual.
nic="eth0"	Indica el nombre de la interfaz a la que se le asignará la dirección IP virtual.
flush_routes="true"	Establece que se elimine las entradas de la tabla de enrutamiento cuando la dirección IP virtual se desasocie del nodo relacionadas con la misma.
arp_sender="iputils_arping"	Indica que se haga uso de arping cuando se asigne la dirección IP virtual a un nodo para notificar de esta manera el cambio de la MAC asociada a la dirección IP en la red.
op monitor interval="10s"	Establece que cada 10 segundo se monitorice el estado del recurso.

3. Crear el recurso con el que se administrará el servidor DHCP en el clúster. Se hará uso de la herramienta pcs y el agente de recurso ocf:heartbeat:dhcpd[68].

```
servidor1#pcs resource create DHCP ocf:heartbeat:dhcpd \
config="/etc/dhcp/dhcpd.conf" \
binary="/usr/sbin/dhcpd" user="root" \
op start interval="0" timeout="60" \
op stop interval="0" timeout="60" \
op monitor interval="10s"
```

Los argumentos usados serán:

resource create	Indica que se va a crear un nuevo recurso en clúster.
DHCP	Nombre que se le asignara al recurso.
ocf:heartbeat:dhcpd	Indica que el agente que con él se hará de uso para el recurso. Entre otras características permite comprobar el estado del recurso realizando una petición al servicio.

config="/etc/dhcp/dhcpd.conf"	Establece el fichero de configuración que se usará al levantar el demonio.
binary="/usr/sbin/dhcpd"	Indica el binario usado para levantar el demonio de DHCP.
user="root"	Establece el usuario con el que se iniciara el demanio de DHCP.
op start interval="0" timeout="60s"	Establece que si el recurso tarda más de un minuto dará como erróneo el intento de arranque. Además, indica que no se espere ante de reintentarlo.
op stop interval="0" timeout="60s"	Establece que si el recurso tarda más de un minuto dará como erróneo el intento de parada. Además, indica que no se espere ante de reintentarlo.
op monitor interval="10s"	Establece que cada 10 segundo se monitorice el estado del recurso.

4. Crear los recursos con el que se administrará el servidor TFTP, gestor de base de datos, servicio de Rsync y Samba en el clúster. Se hará uso de la herramienta pcs.

```
servidor1#pcs resource create TFTP service:tftpd-hpa \
op start interval="0" timeout="60" \
op stop interval="0" timeout="60" \
op monitor interval="10s"
servidor1# pcs resource create BBDD service:mariadb \
op start interval="0" timeout="60" \
op stop interval="0" timeout="60" \
op monitor interval="10s"
servidor1#pcs resource create Rsync service:rsync.service\
op start interval="0" timeout="60" \
op stop interval="0" timeout="60" \
op monitor interval="10s"
servidor1#pcs resource create Samba service:smbd \
op start interval="0" timeout="60" \
op stop interval="0" timeout="60" \
op monitor interval="10s"
servidor1#pcs resource create Samba-NMBD service:nmbd \
op start interval="0" timeout="60" \
op stop interval="0" timeout="60" \
op monitor interval="10s"
```

Los argumentos usados serán:

resource create	Indica que se va a crear un nuevo recurso en clúster.
Y	Nombre que se le asignara al recurso.
service:X	Establece que el clúster administrará el servicio haciendo mediante el uso de la herramienta service[69].
op start interval="0" timeout="60s"	Establece que si el recurso tarda más de un minuto dará como erróneo el intento de arranque. Además, indica que no se espere ante de reintentarlo.
op stop interval="0" timeout="60s"	Establece que si el recurso tarda más de un minuto dará como erróneo el intento de parada. Además, indica que no se espere ante de reintentarlo.
op monitor interval="10s"	Establece que cada 10 segundo se monitorice el estado del recurso.

5. Crear el recurso con el que se administrará el servidor Apache. Se hará uso de la herramienta pcs y el agente de recurso ocf:heartbeat:apache[70].

```
servidor1#pcs resource create Apache ocf:heartbeat:apache  \
configfile=/etc/apache2/apache2.conf \
httpd=/usr/sbin/apache2 \
statusurl="http://localhost/server-status" \
op monitor interval="10s"
```

Los argumentos usados serán:

resource create	Indica que se va a crear un nuevo recurso en clúster.
Apache	Nombre que se le asignara al recurso.
ocf:heartbeat:apache	Indica que el agente que con él se hará de uso para el recurso. Entre otras características permite comprobar el estado del recurso consultara la página configurada para ello.
configfile=/etc/apache2/apache2.conf	Indica el fichero de configuración que se usará al levantar el servidor.
httpd=/usr/sbin/apache2	Indica el binario usado para levantar el servidor web Apache.
statusurl="http://localhost/server-status"	Estable la URL a consultar para comprobar la salud del recurso en el nodo donde se está ejecutando.
op monitor interval="10s"	Establece que cada 10 segundo se monitorice el estado del recurso.

6. Crear el recurso con el que se administrará el servicio de opengnsys. Se hará uso de la herramienta pcs.

```
servidor1#pcs resource create Opengnsys lsb:opengnsys \
op start interval="0" timeout="60" \
op stop interval="0" timeout="60" \
op monitor interval="10s"
```

Los argumentos usados serán:

resource create	Indica que se va a crear un nuevo recurso en clúster.
Opengnsys	Nombre que se le asignara al recurso.
lsb:opengnsys	Indica que se administrará mediante la ejecución del script /etc/init.d/opengnsys con los argumentos start, status y stop.
op start interval="0" timeout="60s"	Establece que si el recurso tarda más de un minuto dará como erróneo el intento de arranque. Además, indica que no se espere ante de reintentarlo.
op stop interval="0" timeout="60s"	Establece que si el recurso tarda más de un minuto dará como erróneo el intento de parada. Además, indica que no se espere ante de reintentarlo.
op monitor interval="10s"	Establece que cada 10 segundo se monitorice el estado del recurso.

7. Crear los recursos con los cuales se realizará la redirección de una petición a un recurso que se encuentra corriendo en el otro nodo. Se hará uso de la herramienta pcs y el agente de recurso ocf:heartbeat:anything.

```
servidor1#pcs resource create iptablesTFTP ocf:heartbeat:anything\
binfile="/opt/scriptHA/scriptReglaIptables.sh" \
cmdline_options="-p '69/udp' -r TFTP"\
logfile=/opt/scriptHA/logs/iptablesTFTP \
op start interval="0" timeout="60s" \
op stop interval="0" timeout="60s" \
op monitor interval="30s" timeout="60s"
servidor1#pcs resource create iptablesDHCP ocf:heartbeat:anything\
binfile="/opt/scriptHA/scriptReglaIptables.sh" \
cmdline_options="-p '67:68/udp' -r DHCP" \
logfile=/opt/scriptHA/logs/ iptablesDHCP\
op start interval="0" timeout="60s" \
op stop interval="0" timeout="60s" \
op monitor interval="30s" timeout="60s"
servidor1#pcs resource create iptablesBBDD ocf:heartbeat:anything\
binfile="/opt/scriptHA/scriptReglaIptables.sh" \
cmdline_options="-p '3306/tcp' -r BBDD"  \
logfile=/opt/scriptHA/logs/iptablesBBDD \
op start interval="0" timeout="60s" \
op stop interval="0" timeout="60s" \
op monitor interval="30s" timeout="60s"
```

Los argumentos usados serán:

resource create	Indica que se va a crear un nuevo recurso en clúster.
X	Nombre que se le asignara al recurso.
binfile=/opt/scriptHA/scriptReglaIptables.sh	Indica la ruta absoluta del script que será ejecutado con el recurso.
cmdline_options="Y"	Incida los argumentos usado en la ejecución del script especificado en el argumento binfile.
logfile=/opt/scriptHA/logs/Z	Indica donde se almacenará la salida estándar generada en la ejecución del script especificado en el argumento binfile.
ocf:heartbeat:anything	Indica que se hará uso del agente de recurso anything que permite realizar la configuración para que el clúster administre cual binario o script.
op start interval="0" timeout="60s"	Establece que si el recurso tarda más de un minuto dará como erróneo el intento de arranque. Además, indica que no se espere ante de reintentarlo.
op stop interval="0" timeout="60s"	Establece que si el recurso tarda más de un minuto dará como erróneo el intento de parada. Además, indica que no se espere ante de reintentarlo.
op monitor interval="30s" timeout="60s"	Establece que cada 30 segundo se monitorice el estado del recurso y si la operación de monitorización tardará mas de un minuto dará como erróneo el intento de arranque.

8. Crear los recursos con los cuales administrarán los dispositivos de bloques administrado con drbd y que se encargar de cambiar el rol de un nodo entre primario o maestro y secundario o esclavo. Se hará uso de la herramienta pcs y el agente de recurso ocf:linbit:drbd[71].

```
servidor1#pcs resource create DrbdImagenes ocf:linbit:drbd\
drbd_resource=imagenes promotable promoted-max=1\
promoted-node-max=1 clone-max=2 clone-node-max=1 notify=true
servidor1#pcs resource create DrbdTFTP ocf:linbit:drbd\
drbd_resource=tftp promotable promoted-max=1\
promoted-node-max=1 clone-max=2 clone-node-max=1 notify=true
servidor1#pcs resource create DrbdBBDD ocf:linbit:drbd\
drbd_resource=basedatos promotable promoted-max=1\
promoted-node-max=1 clone-max=2 clone-node-max=1 notify=true
```

Los argumentos usados serán:

resource create	Indica que se va a crear un nuevo recurso en clúster.
X	Nombre que se le asignara al recurso.
drbd_resource=Y	Indica el nombre del recurso de drbd que será administrado por el recurso del clúster.
promotable	Indica que el recurso será clonado y existirá varias instancias del mismo recurso, con la característica añadida de que existirán instancia que tendrán el rol de maestro y otro de esclavo
promoted-max=1	Indica el número de máximo de instancia con el rol de maestro.
promoted-node-max=1	Indica el número de máximo de instancia con el rol de maestro por nodo.
clone-max=2	Indica el número máximo de instancia del recurso puede encontrarse en clúster.
clone-node-max=1	Indica el número máximo de instancia del recurso se puede encontrar en el mismo nodo.
notify=true	Activa la notificación del estado actual de una instancia al resto de la instancia y de esta manera poder generar cambios como el paso de una instancia del rol de esclavo a rol de maestro.

9. Crear los recursos con realizarán el montaje de los dispositivos de bloques. Se hará uso de la herramienta pcs y el agente de recurso ocf:heartbeat:Filesystem[72].

```
servidor1#pcs resource create DrbdImagenesFS\
ocf:heartbeat:Filesystem device="/dev/drbd1" \
directory="/opt/opengnsys/images/" fstype="ext3"
servidor1#pcs resource create DrbdBbddFS\
ocf:heartbeat:Filesystem device="/dev/drbd2" \
directory="/var/lib/mysql/" fstype="ext3"
servidor1#pcs resource create DrbdTFTPFS\
ocf:heartbeat:Filesystem device="/dev/ drbd3" \
directory=" /srv/tftp/ " fstype="ext3"
```

Los argumentos usados serán:

resource create	Indica que se va a crear un nuevo recurso en clúster.
X	Nombre que se le asignara al recurso.
ocf:heartbeat:Filesystem	Indica que el agente que con él se hará de uso para el recurso. Entre otras características comprobará que el sistema de fichero se encuentra montado correctamente y accesible.
device="/dev/drbdY"	Indica el dispositivo que se montará.
directory="Z"	Indica la ruta absoluta donde se realizará el montaje del dispositivo.
fstype="ext3"	Especifica el tipo de montaje a montar.

10. Crear los grupos de los recursos que deben ejecutarse en el mismo nodo. Se hará uso de la herramienta pcs.

```
servidor1#pcs resource group add OPENGNSYSgrupo PublicaIP \
Apache Opengnsys Samba-NMBD Samba Samba-NMBD Rsync DrbdImagenesFS
servidor1#pcs resource group add TFTPgrupo DrbdTFTPFS TFTP
servidor1#pcs resource group add BBDDgrupo DrbdBbddFS BBDD
```

Los argumentos usados serán:

resource group add	Indica que se va a crear un nuevo grupo de recurso en el clúster.
Xgrupo	Nombre que se le asignara al grupo de recurso.
Y…Z	Especifica todos los recursos que formará el grupo.

11. Configurar los recursos para que en caso de fallar cuatro veces el recurso este cambe de nodo y que el contador de fallo se reinicie cada un minuto. Se hará uso de la herramienta pcs.

```
servidor1#pcs resource meta DHCP migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta TFTP migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta BBDD migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta Rsync migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta Samba migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta Samba-NMBD migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta Apache migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta Opengnsys migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta iptablesDHCP migration-threshold=4 \
failure-timeout=60s
```

```
servidor1#pcs resource meta iptablesTFTP migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta iptablesBBDD migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta DrbdImagenes migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta DrbdImagenesFS migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta DrbdTFTP migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta DrbdTFTPFS migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta DrbdBbdd migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta DrbdBbddFS migration-threshold=4 \
failure-timeout=60ss
servidor1#pcs resource meta OPENGNSYSgrupo migration-threshold=4 \
failure-timeout=60s
servidor1#pcs resource meta BBDDgrupo migration-threshold=4 \
failure-timeout=60
servidor1#pcs resource meta TFTPgrupo migration-threshold=4 \
failure-timeout=60s
```

Los argumentos usados serán:

resource meta	Indica que se va a modificar un parámetro de la administración de un recurso.
X	Nombre del recurso a configurar.
migration-threshold=4	Especifica que se mueva el recurso al otro nodo después de que ocurran 4 fallos.
failure-timeout=60s	Especifica que se reinicien el contador de fallo pasado un minuto.

12. Establecer las restricciones en cuanto a la ubicación de los recursos respecto a otros recursos, pueden ser que dos recursos no se ejecuten nunca en el mismo nodo o que siempre deba ejecutarse en el mismo nodo. Se hará uso de la herramienta pcs.

```
servidor1#pcs constraint colocation add iptablesBBDD with BBDD \
score=-INFINITY
servidor1#pcs constraint colocation add iptablesTFTP with TFTP \
score=-INFINITY
servidor1#pcs constraint colocation add iptablesDHCP with DHCP \
score=-INFINITY
servidor1#pcs constraint colocation add iptablesBBDD with BBDD \
score=-INFINITY
servidor1#pcs constraint colocation add OPENGNSYSgrupo \
with DrbdImagenes-clone score=INFINITY with-rsc-role=Master
servidor1#pcs constraint colocation add TFTPgrupo \
with DrbdTFTP-clone score=INFINITY with-rsc-role=Master
servidor1#pcs constraint colocation add BBDDgrupo \
with DrbdBbdd-clone score=INFINITY with-rsc-role=Master
```

Los argumentos usados serán:

constraint colocation add	Indica que va a crear una restricción de colocación de los recursos.
X with Y	Indica los pares de recursos sobre los cuales se va a aplicar la restricción.
score=Z	Establece el valor de la afinidad entre los recursos. Mientras mayor sea esto indica que tienen más prioridad para ejecutarse en el mismo nodo. El valor INFINITY indica que ejecutar siempre en el mismo nodo. El valor -INFINITY indica que nunca debe ejecutar en el mismo nodo
with-rsc-role=Master	Es usado cuando uno de los recursos se trata de un recurso clonado y con roles de maestro-esclavo para indica que la restricción se aplicara sobre uno de los roles. Esto puede usarse para que un recurso se ejecute siempre en el nodo donde se encuentre el maestro.

13. Establecer las restricciones que indicará el recurso a iniciar tras cambiar el rol de esclavo a maestro un recurso, esto será usado para iniciar el montaje de un dispositivo después de clúster establecido que nodo será el maestro. Se hará uso de la herramienta pcs.

```
servidor1#pcs constraint order promote DrbdImagenes-clone \
then start OPENGNSYSgrupo
servidor1#pcs constraint order promote DrbdTFTP-clone \
then start TFTPgrupo
servidor1#pcs constraint order promote DrbdBbdd-clone \
then start BBDDgrupo
```

Los argumentos usados serán:

constraint order	Indica que va a crear una restricción para indicar el orden en el que se realizara una serie de acciones.
promote X-clone	Indica el recurso cuyo cambio de rol a maestro provocará la siguiente acción.
then start Ygrupo	Indica el recurso que se iniciará tras el cambio de rol.

14. Establecer las preferencias donde se desea que se coloque los diferentes recursos. Se hará uso de la herramienta pcs.

```
servidor1#pcs constraint location DHCP prefers \
servidor1=1000 servidor2=0
servidor1#pcs constraint location TFTP prefers \
servidor1=1000 servidor2=0
servidor1#pcs constraint location BBDD prefers \
servidor1=1000 servidor2=0
servidor1#pcs constraint location iptablesBBDD \
prefers servidor2=1000 servidor1=0
servidor1#pcs constraint location iptablesTFTP \
prefers servidor2=1000 servidor1=0
```

```
servidor1#pcs constraint location iptablesDHCP \
prefers servidor2=1000 servidor1=0
servidor1#pcs constraint location DrbdImagenes-clone \
prefers servidor1=1000 servidor2=0
servidor1#pcs constraint location DrbdImagenesFS \
prefers servidor1=1000 servidor2=0
servidor1#pcs constraint location DrbdTFTP-clone \
prefers servidor1=1000 servidor2=0
servidor1#pcs constraint location DrbdTFTPFS prefers \
servidor1=1000 servidor2=0
servidor1#pcs constraint location DrbdBbdd-clone \
prefers servidor1=1000 servidor2=0
servidor1#pcs constraint location DrbdBbddFS prefers \
servidor1=1000 servidor2=0
servidor1#pcs constraint location OPENGNSYSgrupo prefers \
servidor1=1000 servidor2=0
servidor1#pcs constraint location TFTPgrupo prefers \
servidor1=1000 servidor2=0
servidor1#pcs constraint location BBDDgrupo prefers \
servidor1=1000 servidor2=0
```

Los argumentos usados serán:

pcs constraint location	Indica que va a crear una restricción que indicará donde se prefiere o se quiere evitar que se coloque un recurso.
X	Indica el recurso sobre el que se aplicará la restricción.
prefers	Indica que la regla será de preferencia.
servidorY=Z	Indica el nodo y el nivel de preferencia de este. A mayor valor para Z el nodo tendrá preferencia a la hora de que el recurso se establezca en ese nodo.

15. Establecer la prioridad de los recursos, esto permite que si no se pueden ejecutar todos los recursos por algunas restricciones se elegirán los recursos ejecutar en a esta propiedad. Se hará uso de la herramienta pcs.

```
servidor1#pcs resource meta DrbdImagenes priority=1000
servidor1#pcs resource meta DrbdTFTP priority=1000
servidor1#pcs resource meta DrbdBbdd priority=1000
servidor1#pcs resource meta OPENGNSYS priority=100
servidor1#pcs resource meta BBDDgrupo priority=100
servidor1#pcs resource meta TFTPgrupo priority=100
servidor1#pcs resource meta iptablesMysql priority=1
servidor1#pcs resource meta iptablesTFTP priority=1
servidor1#pcs resource meta iptablesDHCP priority=1
```

Los argumentos usados serán:

resource meta	Indica que se va a modificar un parámetro de la administración de un recurso.
X	Nombre del recurso a configurar.
priority=Y	Indica la prioridad del recurso, mientras mayor sea el valor de Y más prioritario será.

16. Habilitar el seguimiento de conexiones TFTP cuando se redirija el tráfico entre nodos, para ello añadiremos al kernel de Linux varios módulos mediante el uso de la herramienta modprobe.

```
servidor1#modprobe nf_conntrack_tftp
servidor1#modprobe nf_nat_tftp
servidor2#modprobe nf_conntrack_tftp
servidor2#modprobe nf_nat_tftp
```

El argumento usado será:

nf_conntrack_tftp o nf_nat_tftp	Indica el módulo que se desea activar.

17. Configurar para que se cargue automáticamente los módulos necesarios para el redireccionamiento de conexiones TFTP de drbd al kernel de Linux al iniciar el sistema operativo. Para ello se creará un fichero que indicará que modulo se debe a cargar al iniciar al introducirlo en el directorio /etc/modules-load.d/.

```
servidor1#echo nf_conntrack_tftp > \
/etc/modules-load.d/nf_conntrack_tftp.conf
servidor1#echo nf_nat_tftp > /etc/modules-load.d/nf_nat_tftp.conf
servidor2#echo nf_conntrack_tftp > \
/etc/modules-load.d/nf_conntrack_tftp.conf
Servidor2#echo nf_nat_tftp > /etc/modules-load.d/nf_nat_tftp.conf
```

Los argumentos usados serán:

nf_conntrack_tftp o nf_nat_tftp	Indica el módulo que se desea activar.
/etc/modules-load.d/drbd.conf o /etc/modules-load.d/nf_nat_tftp.conf	Crea un fichero que indica el módulo a cargar al iniciar el sistema.

18. Configurar la capacidad de reenvío y el ayudante de seguimiento de conexiones para iptables. Para ello modificaremos el fichero de configuración /etc/sysctl.conf

```
servidor1#echo net.netfilter.nf_conntrack_helper=1 \
 >> /etc/sysctl.conf
servidor1#echo net.ipv4.ip_forward=1 >> /etc/sysctl.conf
servidor2#echo net.netfilter.nf_conntrack_helper=1 \
 >> /etc/sysctl.conf
Servidor2#echo net.ipv4.ip_forward=1 >> /etc/sysctl.conf
```

Para activar estas características del kernel sin reiniciar sería necesario ejecutar el comando: sysctl -p.

19. Deshabilitamos el inicio automático de todos los servicios para los que se crearon los recursos, ya que el clúster será el encargo de iniciarlo o detenerlo. Para ello se usará de la herramienta systemctl.

```
servidor1#systemctl disable isc-dhcp-server.service
servidor1#systemctl disable tftpd-hpa
servidor1#systemctl disable mariadb
servidor1#systemctl disable rsync
servidor1#systemctl disable smbd
servidor1#systemctl disable nmbd
servidor1#systemctl disable apache
servidor1#systemctl disable opengnsys
servidor2#systemctl disable isc-dhcp-server.service
servidor2#systemctl disable tftpd-hpa
servidor2#systemctl disable mariadb
servidor2#systemctl disable rsync
servidor2#systemctl disable smbd
servidor2#systemctl disable nmbd
servidor2#systemctl disable apache
servidor2#systemctl disable opengnsys
```

Los argumentos usados serán:

disable	Establece que el servicio se deshabilita el autoarranque al iniciar el sistema.
…	Indica el nombre del servicio para el que ese deshabilitara el autoarranque.

3.3.3 Líneas de continuación

Una de las posibles de líneas de continuación puede ser aplicar soluciones basada en contenedores y su orquestación mediante Docker Swarm o Kubernertes.

Otra posible línea continuación puede ser la configuración y uso de sistema de ficheros como ocfs2 o gfs2 sobre los dispositivos administrado con drbd, ya que permite la escritura y montaje en varios nodos. En las primeras pruebas realizada para ver su viabilidad para la solución adoptada resulto inestable a la hora de realizar múltiples escrituras desde los dos nodos.

Por último, otra línea de continuación es el cambio de aquellos recursos que hacen uso de los agentes de recursos service o lsb por otros que permitan monitorizar el estado real de recurso y no únicamente el estado del PID asociado al recurso.

4 DESPLIEGUE AUTOMATIZADO Y GESTIÓN REMOTA DE EQUIPOS

En este capítulo se detallará una serie de modificaciones en servicios usado en el Laboratorio de Telemática y el desarrollo de funcionalidades orientada a la gestión de los equipos del laboratorio. La intención de estas es corregir algunos errores encontrado, añadir funcionalidades de utilidad que no existe en la actualidad, permitir el uso de estos equipos desde el exterior del laboratorio y limitar el acceso del usuario en alguno de los casos.

4.1 Arranque de máquina virtual de manera automática

Se detallará se explicará una serie de procesos que permiten iniciar una máquina virtual basado en el uso de la tecnología VMware o VirtualBox de manera que el usuario únicamente pueda hacer uso de la máquina virtual y no acceder al sistema si cierra la maquina o la minimiza.

4.1.1 Situación actual y problemática

En el laboratorio de Telemática se posee un sistema que es usado por los profesores y que se encuentra conectado al proyector del laboratorio. En esta se aloja una máquina virtual de Windows 7 basada en la tecnología VMware que permite a los profesores un uso más amigable de cara a facilitar la impartición de las clases y uso del proyector para presentar diapositivas u otro tipo contenido.

Se quiere evitar que el usuario pueda acceder al sistema que aloja la máquina virtual si la cerrará, apagará o minimizará o mediante una alguna opción del gestor de máquina virtual, para ello se ha ideado una serie de script que serán lanzados por un servicio que se encargarán de lanzar la máquina virtual en un entorno grafico limitado o lanzar el cliente de RPD o VNC para conectarse a la máquina virtual.

Las posibilidades en que se han pensado son:

- Hacer uso de un usuario con permisos reducidos para acceder a la máquina virtual y entorno grafico reducido donde se lanzará la máquina virtual con su lanzador propio de la máquina virtual ya sea VMware o VirtualBox.
- Hacer uso de un cliente de RDP para hacer a las máquinas virtuales de VirtualBox o un cliente VNC para las máquinas virtuales de VMware.

4.1.2 Solución elegida

En las diferentes soluciones que se hará uso de un servicio administrado y lanzado por systemd[73]. Este servicio permitirá que se autoinicie la máquina virtual y el cliente RPD o VNC o el entorno grafico reducido con el gestor de la máquina virtual.

Para crear el servicio será necesario seguir los siguientes pasos:

1. Crear el fichero con el nombre virtual lanzarMaquina.service en la carpeta /usr/lib/systemd/system/ con el siguiente contenido.

```
[Unit]
Description=Lanzadera de maquina virtual
Before=vmware.service

[Service]
ExecStart=/usr/bin/xinit "/opt/servicioMaquina/lanzarMaquina.sh" -
- :1
ExecReload=/usr/bin/xinit  "/opt/servicioMaquina/lanzarMaquina.sh"
-- :1
Restart=always

[Install]
WantedBy=vmware.service
WantedBy=graphical.target
```

Los campos se usado para el servicio son:

Description=Lanzadera de maquina virtual	Establece una descripción corta del servicio.
Before=vmware.service	Indica que servicio debe iniciarse antes de poder iniciar este. En este caso se indica que se debe iniciar antes el servicio de vmware.service y por lo tanto este servicio estaría pensado para usar máquinas virtuales basada en VMware, si se quisiera hacer uso de maquina virtuales basada en VirtualBox el campo debería ser vboxdrv.service.
ExecStart=comando	Indica el comanda a ejecutar cuando se inicia el servicio.
ExecReload=comando	Indica el comanda a ejecutar cuando se reinicie el servicio.
Restart=always	Establece que el servicio se reinicie siempre que la ejecución del comando finalice.
WantedBy=vmware.service	Establece la dependencia otro servicio. En este caso el servicio es vmware.service porque estaría pensado para usar máquinas virtuales basada en VMware. Si se quisiera hacer uso de máquinas virtuales basado en VirtualBox el valor debería ser vboxdrv.service.
WantedBy=grapical.target	Indica que no se inicie el servicio si el sistema no se encuentra en el nivel de ejecución en el cual se inicia el entorno gráfico.

Para consultar más información acerca de los campos del posibles del archivo del servicio se puede consultar el manual de systemd.service[74].

Se utilizará a herramienta xinit[75] ya que permite el inicio de un gestor de ventana que se usará para establecer en el entorno grafico reducido o el cliente de RDP o VNC. La ejecución que se realizará con el servicio será:

```
/usr/bin/xinit "/opt/servicioMaquina/lanzarMaquina.sh" -- :1
```

Los argumentos usados serán:

"/opt/servicioMaquina/lanzarMaquina.sh"	Indica el script que se ejecutará tras iniciar el gestor de ventana
-- :1	Establece que se el gestor de ventana se inicie el gestor de ventana en la primera consola que esté disponible.

Al iniciar el gestor ventana el sistema cambiará a la consola donde se inicie. Para cambiar entre consola del sistema se usará la combinación de teclas [Ctrl] + [Alt] + [Fx]. Siendo X un número entre 1 y 12.

2. Establecer los permisos adecuado al fichero generado para ello se hará uso de la herramienta chmod.

```
chmod 644 /usr/lib/systemd/system/lanzarMaquina.service
```

Los argumentos usados serán:

644	Estable que los permisos para que todos los usuarios puedan leer el fichero y que solo el propietario pueda modificarlo.
/usr/lib/systemd/system/lanzarMaquina.service	Indica el fichero al cual se cambiará los permisos

3. Establecer el grupo y usuario propietario adecuado al fichero generado para ello se hará uso de la herramienta chown.

```
chown root:root /usr/lib/systemd/system/lanzarMaquina.service
```

Los argumentos usados serán:

root:root	Estable que tanto el usuario propietario como el grupo sea root. El primero indica el usuario y el segundo indica el grupo.
/usr/lib/systemd/system/lanzarMaquina.service	Indica el fichero al cual se cambiará los permisos

4. Habilitar el servicio para que se inicie automáticamente al iniciar el sistema para ello se hará uso de la herramienta systemctl.

```
systemctl enable lanzarMaquina.service
```

Los argumentos usados serán:

enable	Establece que el servicio que inicie con el arranque del sistema.
lanzarMaquina.service	Indica el servicio que se iniciará automáticamente.

5. Iniciar el servicio mediante el uso de la herramienta systemctl.

```
systemctl start lanzarMaquina.service
```

Los argumentos usados serán:

start	Indica que se incide un servicio.
lanzarMaquina.service	Indica el servicio que se iniciará.

4.1.2.1 Uso del entorno grafico reducido y lanzador de máquina virtual

Para esta solución ideada se necesitará tener instalado el gestor de ventana i3[76], [77], para realizar la instalación dependerá de la distribución del que se haga uso.

Para distribuciones como Debian o Ubuntu se puede instalar directamente desde el repositorio con el uso de la herramienta apt.

```
apt install -y i3
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
i3	Es el nombre del paquete a instalar.

Para distribuciones para distribuciones como Centos y similares antes de realizar la instalación será necesario habilitar el repositorio epel-release donde se encuentra el paquete i3. Para haremos uso de la herramienta yum[77]

```
yum install -y epel-release
yum update -y
yum install -y i3
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
epel-release	Es el nombre del paquete a instalar que permitirá acceder al repositorio "Extra Packages for Enterprise Linux".

```
yum update -y
```

Los argumentos usados serán:

update	Indica que se actualicen los paquetes instalada a la última versión disponible. Esto es necesario realizarlo para poder actualizar los paquetes disponibles para tener en cuenta los paquetes en el repositorio "Extra Packages for Enterprise Linux".
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de actualización y de esta manera realizar la instalación de forma no interactiva.

```
yum install -y i3
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
i3	Es el nombre del paquete a instalar.

Tambien será necesario posee un usuario cuyo permiso sean restringido y el cual solo pueda acceder al directorio donde se aloje la máquina virtual y contendido del mismo. Para para el ejemplo del script que se expondrá a continuación se asumirá que este usuario es usuarioMaquinaVirtual.

El contenido el script /opt/servicioMaquina/lanzarMaquina.sh si usamos maquina virtuales basada en VirtualBox será para usar:

```
#!/bin/sh

cambioPermiso(){
CARPETA=$(dirname "$1")
chmod -R o+wr "$CARPETA"
 while [ "$CARPETA" != "/" ]
   do
    chmod o+x "$CARPETA"
    CARPETA=$(dirname "$CARPETA")
   done
}

VMFile="/var/share/MaquinaVirtual/archivoMaquina.vbox"
VMName=$(basename $VMFile | cut -d"." -f1)
i3Cofig="/opt/servicioMaquina/i3config"
i3 -c $i3Cofig &
cambioPermiso "$VMFile"
su usuarioMaquinaVirtual -c "vboxmanage list vms " | grep -q
"\"$VMName\""
if [ "$?" = "1"]; then
 su usuarioMaquinaVirtual -c "VBoxManage registervm \"$VMFile\""
```

```
 fi
 su usuarioMaquinaVirtual -c "virtualboxvm --startvm \"$VMName\" --
fullscreen"
 exit
```

El contenido el script /opt/servicioMaquina/lanzarMaquina.sh si usamos maquina virtuales basada en VMware será para usar:

```
 #!/bin/sh

 cambioPermiso(){
 CARPETA=$(dirname "$1")
 chmod -R o+wr "$CARPETA"
  while [ "$CARPETA" != "/" ]
    do
     chmod o+x "$CARPETA"
     CARPETA=$(dirname "$CARPETA")
    done
 }

 VMFile="/var/share/MaquinaVirtual/archivoMaquina.vmx"
 i3Cofig="/opt/servicioMaquina/i3config"
 i3 -c $i3Cofig &
 cambioPermiso "$VMFile"
 su usuarioMaquinaVirtual -c "vmplayer -X \"$VMFile\""
 exit
```

Las partes en la que se dividen ambos scripts son:

1. Declaración de la ubicación del fichero de la máquina virtual mediante la variable VMFile, en esta variable es necesario detallara la ruta completa.

```
 VMFile="/var/share/MaquinaVirtual/archivoMaquina.vbox"
```

Declaración de la ubicación del fichero de configuración del gestor de ventana i3 mediante la variable i3Cofig, en el

2. ANEXO H: Arranque de máquina virtual de manera automática se adjunta el contenido de un fichero de configuración con la configuración mínima para que se ejecute el lanzador de la maquina correctamente.

```
i3Cofig="/opt/servicioMaquina/i3config"
```

3. Se lanza el gestor de ventana i3 con la configuración reducida.

```
i3 -c $i3Cofig &
```

Los argumentos usados serán:

-c $i3Cofig	Selecciona la configuración para el gestor de ventana.
&	Para ejecutarlo en segundo plano y permitir continuar con el script.

4. Se lanza la función cambioPermiso con la variable VMFile como argumento. Esta función cambiará los permisos de escritura, lectura y ejecución para otros usuarios para que la máquina virtual pueda ser usada por el usuario con permiso reducido. En esta función se recorrerá los directorios superiores para asignarles permisos de ejecución para poder acceder a la carpeta que contiene la máquina virtual y para los archivos que conforma la máquina virtual asignará permisos de lectura y escritura para otros usuarios. Para ello se usará las herramientas chmod y dirname[78].

```
cambioPermiso "$VMFile"
```

5. Ejecución del comando con el usuario con permiso reducido para iniciar la máquina virtual para ello se hará uso de la herramienta su[79].

```
su usuarioMaquinaVirtual -c "comandoMaquina"
```

Los argumentos usados serán:

usuarioMaquinaVirtual	Indica el usuario con el cual se ejecutará la sentencia indicada.
-c "comandoMaquina"	Indica la sentencia a ejecutar.

La sentencia comandoMaquina variará en función de la tecnología en la que esté basada la máquina virtual:

- Para aquella basada en VMware se hará uso de la herramienta vmplayer cuyos posibles argumentos son los mismo que vmware[80].

```
vmplayer -X \"$VMFile\"
```

Los argumentos usados serán:

-X	Indica que la máquina virtual inicie en pantalla completa.
\"$VMFile\"	Hace referencia la ubicación del fichero que define la máquina virtual.

- Para aquella basada en VirtualBox se hará uso de la herramienta virtualboxvm[81]. y se iniciará a partir del nombre de la máquina virtual.

```
virtualboxvm --startvm \"$VMName\" --fullscreen
```

Los argumentos usados serán:

--startvm	Especifica que se inicie una máquina virtual.
\"$VMName\"	Hace referencia al nombre de la máquina virtual a iniciar.
--fullscreen	Indica que la máquina virtual inicie en pantalla completa.

Para estas máquinas es necesario que la maquina este importada para el usuario que lanzará la máquina virtual.

Para comprobar que la máquina que la maquina ya ha sido registrada se usara la herramienta vboxmanage[82].

```
vboxmanage list vms
```

Los argumentos usados serán:

list vms	Especifica que se va a listar las máquinas virtuales importadas.

De no aparecer como el resultado de la ejecución del comando anterior sería necesario registrar la máquina virtual a partir del fichero y para ello se hará uso de la herramienta vboxmanage.

```
vboxmanage registervm "$VMFile"
```

Los argumentos usados serán:

registervm	Especifica que se va a registrar una nueva máquina a partir de un fichero que lo define.
"$VMFile"	Hace referencia la ubicación del fichero que define la máquina virtual.

4.1.2.2 Uso de cliente VNC o RDP

Esta solución ideada se divide en uso de un cliente VNC o RDP debido a que las máquinas virtuales basada en VMware solo permiten remotamente conectarse mediante el protocolo VNC y en oposición aquellas basadas en VirtualBox solo permiten el conectarse remotamente mediante el protocolo RDP.

Para esta solución deberá está configurada las máquinas virtuales para permitir la conexión mediante VNC o RPD, en el

ANEXO H: Arranque de máquina virtual de manera automática se detallará como realizar la configuración. Se recomienda que se tenga configurado la conexión automática de dispositivo extraíbles a la máquina virtual ya que no será posible acceder a los menús de la máquina para realizar la conexión.

4.1.2.2.1 Uso de cliente VNC para máquinas virtuales basada en VMware

Para la máquina virtuales basadas en VMware es posible su control remoto mediante el protocolo VNC, por ello se hará uso del cliente VNC vncviewer[83].

Para realizar la instalación vncviewer en distribuciones como Debian o Ubuntu se puede instalar directamente desde el repositorio con el uso del comando apt.

```
apt install -y tigervnc-viewer tigervnc-common
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
tigervnc-viewer tigervnc-common	Es el nombre de los dos paquetes a instalar.

Para realizar la instalación vncviewer en distribuciones como Centos y similares se puede instalar directamente desde el repositorio con el uso de la herramienta yum.

```
yum install -y tigervnc
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
tigervnc	Es el nombre del paquete a instalar.

El contenido el script /opt/servicioMaquina/lanzarMaquina.sh será:

```
#!/bin/sh
PUERTO_PROTOCOLO="5999"
IP_PROTOCOLO="localhost"
VMFile="/var/share/MaquinaVirtual/archivoMaquina.vmx"
vmplayer "$VMFile" &
PID_VM="$!"

timeout 30s sh -c "while ! nc -z ${IP_PROTOCOLO} ${PUERTO_PROTOCOLO}
; do sleep 1; done"
nc -z ${IP_PROTOCOLO} ${PUERTO_PROTOCOLO}
```

```
if [ "$?" == "0" ]; then
 vncviewer -FullScreen "${IP_PROTOCOLO}:${PUERTO_PROTOCOLO}"
fi

kill -9 "$PID_VM"
exit
```

El script se divide en cuatros partes diferenciadas:

- Activación la máquina virtual y guardado el id del proceso.

Para aquella basada en VirtualBox se hará uso de la herramienta virtualboxvm y se iniciará a partir del nombre de la máquina virtual.

```
virtualboxvm --startvm "$VMName"
```

Los argumentos usados serán:

--startvm	Especifica que se inicie una máquina virtual.
"$VMName"	Hace referencia al nombre de la máquina virtual a iniciar.

Para estas máquinas es necesario que la maquina este importada para el usuario que lanzará la máquina virtual.

Para comprobar que la máquina que la maquina ya ha sido registrada se usara la herramienta vboxmanage.

```
vboxmanage list vms
```

Los argumentos usados serán:

list vms	Especifica que se va a listar las máquinas virtuales importadas.

De no aparecer como el resultado de la ejecución del comando anterior sería necesario registrar la máquina virtual a partir del fichero y para ello se hará uso de la herramienta vboxmanage.

```
vboxmanage registervm "$VMFile"
```

Los argumentos usados serán:

registervm	Especifica que se va a registrar una nueva máquina a partir de un fichero que lo define.
"$VMFile"	Hace referencia la ubicación del fichero que define la máquina virtual.

Y luego se guardará en la variable PID_VM el identificador del proceso mediante el comando.

```
PID_VM="$!"
```

- Espera para que la maquina esta activa y el servidor VNC levantado.

Para realizar la espera se hará uso de un bucle que comprobará mediante el uso de la herramienta nc[84] que el puerto VNC ya se encuentra accesible. En este bucle consultará el puerto cada segundo como un máximo de 30s. El bucle usado es el siguiente.

```
 timeout 30s sh -c "while ! nc -z ${IP_PROTOCOLO} ${PUERTO_PROTOCOLO}
; do sleep 1; done"
```

Este bucle se divide dos partes:

  - El uso de la herramienta timeout[85] con la cual se establece un temporizador para que no se quede en bucle por si no se levantará el servidor VNC.

```
 timeout 30s sh -c "commando"
```

Los argumentos usados serán:

30s	Establece el tiempo del temporizador.
sh -c "commando"	Establece el comando por el cual se esperará a que finalice o se forzará el fin pasa el tiempo especificado.

  - El uso de la herramienta nc en un bucle while donde comprobará el puerto de VNC es accesible y de no serlo se esperá un segundo mediante la herramienta sleep[86].

```
 while ! nc -z ${IP_PROTOCOLO} ${PUERTO_PROTOCOLO} ; do sleep 1; done
```

El bucle while de forma expandida es

```
 while ! nc -z ${IP_PROTOCOLO} ${PUERTO_PROTOCOLO}
 do
  sleep 1
done
```

En el bucle la condición de entrada viene dada por la negación el resultado de la ejecución de la herramienta nc que dará como resultado 1 si el puerto no es accesible y 0 si es accesible.

Los argumentos usados con herramienta serán:

-z	Establece que haga la búsqueda de demonio escuchando en un puerto sin enviarle datos.
${IP_PROTOCOLO}	Indica la dirección donde se ubica el puerto a consultar. Para este caso siempre será localhost al levantarse la máquina virtual en local.
${PUERTO_PROTOCOLO}	Indica el puesto a consulta

- Conexión con el cliente VNC a la máquina virtual.

Para relazar la conexión mediante VNC se hará uso de la herramienta vncviewer.

```
 vncviewer -FullScreen "${IP_PROTOCOLO}:${PUERTO_PROTOCOLO}"
```

Los argumentos usados serán:

-FullScreen	Establece que inicie en pantalla completa.
${IP_PROTOCOLO}	Indica la dirección donde se encuentra la máquina. Para este caso siempre será localhost al levantarse la máquina virtual en local.
${PUERTO_PROTOCOLO}	Indica el puerto donde se encuentra el servidor VNC

En el caso en él se establezca una contraseña para la conexión mediante VNC será necesario añadir un nuevo argumento:

-passwd="/opt/servicioMaquina/passwd"	Indica el archivo que contiene la contraseña cifrada para la conexión VNC. Este archivo se ha de generar con la herramienta vncpasswd[87].

- Cerrado de la máquina virtual.

Como protección en caso de que el usuario cierre el cliente VNC se cerrará la máquina virtual mediante el identificador del proceso y de esta manera asegurar de que la maquina es cerrada. Para ello se hará uso de la herramienta kill[88].

```
kill -9 "$PID_VM"
```

Los argumentos usados serán:

-9	Indica que se envié la señal SIGKILL para terminar inmediatamente el proceso
"$PID_VM"	Identificador del proceso a cerrar.
${PUERTO_PROTOCOLO}	Indica el puerto donde se encuentra el servidor VNC

4.1.2.2.2 Uso de cliente RDP para máquinas virtuales basada en VirtualBox

Para la máquina virtuales basadas en VirtualBox es posible su control remoto mediante el protocolo RDP, por ello se hará uso del cliente RDP freerdp[89].

Para realizar la instalación freerdp en distribuciones como Debian o Ubuntu se puede instalar directamente desde el repositorio con el uso del comando apt

```
apt install -y freerdp2-x11
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.

freerdp2-x11	Es el nombre de los dos paquetes a instalar.

Para realizar la instalación freerdp en distribuciones como Centos y similares se puede instalar directamente desde el repositorio con el uso de la herramienta yum.

```
yum install -y freerdp
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
freerdp	Es el nombre del paquete a instalar.

El contenido el script /opt/servicioMaquina/lanzarMaquina.sh será:

```
#!/bin/sh
PUERTO_PROTOCOLO="5999"
IP_PROTOCOLO="localhost"
VMFile="/var/share/MaquinaVirtual/archivoMaquina.vbox"
VMName=$(basename $VMFile | cut -d"." -f1)
vboxmanage list vms | grep -q "\"$VMName\""
if [ "$?" = "1"]; then
 vboxmanage registervm "$VMFile"
fi
virtualboxvm --startvm "$VMName" &
PID_VM="$!"

timeout 30s sh -c "while ! nc -z ${IP_PROTOCOLO} ${PUERTO_PROTOCOLO}
; do sleep 1; done"
nc -z ${IP_PROTOCOLO} ${PUERTO_PROTOCOLO}
if [ "$?" == "0" ]; then
 xfreerdp  /v:"${IP_PROTOCOLO}:${PUERTO_PROTOCOLO}" /f
fi

kill -9 "$PID_VM"
exit
```

El script se divide en cuatros partes diferenciadas:

- Activación la máquina virtual y guardado el id del proceso.

Para activa la maquina se basada en VMware se hará uso de la herramienta vmplayer cuyos posibles argumentos son los mismo que vmware.

```
vmplayer "$VMFile" &
```

Los argumentos usados serán:

"$VMFile"	Hace referencia la ubicación del fichero que define la máquina virtual.
&	Indica que el proceso se ejecute en segundo plano.

Y luego se guardará en la variable PID_VM el identificador del proceso mediante el comando.

```
PID_VM="$!"
```

- Espera para que la maquina esta activa y el servidor RDP levantado.

Para realizar la espera se hará uso de un bucle que comprobará mediante el uso de la herramienta nc que el puerto RDP ya se encuentra accesible. En este bucle consultará el puerto cada segundo como un máximo de 30s. El bucle usado es el siguiente.

```
timeout 30s sh -c "while ! nc -z ${IP_PROTOCOLO} ${PUERTO_PROTOCOLO}
; do sleep 1; done"
```

Este bucle se divide dos partes:

- El uso de la herramienta timeout con la cual se establece un temporizador para que no se quede en bucle por si no se levantará el servidor VNC.

```
timeout 30s sh -c "commando"
```

Los argumentos usados serán:

30s	Establece el tiempo del temporizador.
sh -c "commando"	Establece el comando por el cual se esperará a que finalice o se forzará el fin pasa el tiempo especificado.

- El uso de la herramienta nc en un bucle while donde comprobará el puerto de VNC es accesible y de no serlo se esperá un segundo mediante la herramienta sleep.

```
while ! nc -z ${IP_PROTOCOLO} ${PUERTO_PROTOCOLO} ; do sleep 1; done
```

El bucle while de forma expandida es

```
while ! nc -z ${IP_PROTOCOLO} ${PUERTO_PROTOCOLO}
do
 sleep 1
done
```

En el bucle la condición de entrada viene dada por la negación el resultado de la ejecución de la herramienta nc que dará como resultado 1 si el puerto no es accesible y 0 si es accesible.

Los argumentos usados con herramienta serán:

-z	Establece que haga la búsqueda de demonio escuchando en un puerto sin enviarle datos.
${IP_PROTOCOLO}	Indica la dirección donde se ubica el puerto a consultar. Para este caso siempre será localhost al levantarse la máquina virtual en local.
${PUERTO_PROTOCOLO}	Indica el puesto a consulta

- Conexión con el cliente RDP a la máquina virtual.

Para relazar la conexión mediante RDP se hará uso de la herramienta freerdp.

```
xfreerdp /v:"${IP_PROTOCOLO}:${PUERTO_PROTOCOLO}" /f
```

Los argumentos usados serán:

/v:	Establece que inicie en pantalla completa.
${IP_PROTOCOLO}	Indica la dirección donde se encuentra la máquina. Para este caso siempre será localhost al levantarse la máquina virtual en local.
${PUERTO_PROTOCOLO}	Indica el puerto donde se encuentra el servidor VNC
/f	Establece que inicie en pantalla completa.

En el caso en él se establezca una contraseña para la conexión mediante VNC será necesario añadir un nuevo argumento:

-passwd="/opt/servicioMaquina/passwd"	Indica el archivo que contiene la contraseña cifrada para la conexión VNC. Este archivo se ha de generar con la herramienta vncpasswd.

- Cerrado de la máquina virtual.

Como protección en caso de que el usuario cierre el cliente VNC se cerrará la máquina virtual mediante el identificador del proceso y de esta manera asegurar de que la maquina es cerrada. Para ello se hará uso de la herramienta kill.

```
kill -9 "$PID_VM"
```

Los argumentos usados serán:

-9	Indica que se envié la señal SIGKILL para terminar inmediatamente el proceso
"$PID_VM"	Identificador del proceso a cerrar.

4.1.3 Líneas de continuación

Una posible línea de continuación estudiar la manera de que una máquina virtual pueda ser iniciada

automáticamente y a la vez un usuario pueda tomar el control desde una sesión para realizar modificaciones sobre la misma.

4.2 Modificaciones y procedimientos de Opengnsys

En esta sección se detallará serie de modificaciones y/o correcciones sobre el software Opengnsys que se han visto necesarias para la mejora del funcionamiento en el laboratorio de Telemática. Tambien se detallará algunos procedimientos de utilidad para el funcionamiento de este.

Opengnsys se trata de un software usado para la gestión todos los equipos del laboratorio, esto va desde el despliegue de los diferentes sistemas operativos y su configuración en cada equipo la gestión remota de los mismos con el fin de poder apagar, encender o ejecutar script.

4.2.1 Situación actual y problemática

En el laboratorio de Telemática actualmente se encuentra desplegada la versión 1.0.6, sobre esta versión se ha detecta una serie de errores debido al arquitectura de red de este y funcionalidades faltantes que facilitan su uso por parte de los profesores y administradores de este.

Antes de enumerar los temas que se abordarán se debe explicar de forma simplificada las características de la red del laboratorio que influyen en el funcionamiento de Opengnsys son:

- Se dispone de dos subredes entre la que se dividen los equipos del laboratorio.
- El servidor donde se encuentra desplegado Opengnsys es quien cumple la función de encaminador y puerta de enlace para todo el laboratorio.
- El servidor dispone de tres interfaces, dos de ellas está dedicada a la conexión con una de las subredes (interfaz eth1 para la subred A e interfaz eth2 para la subred B) y la tercera está conectada al exterior del laboratorio (interfaz eth0).
- La interfaz por defecto asociada a Opengnsys es que se encuentra conectada al exterior del laboratorio (interfaz eth0).

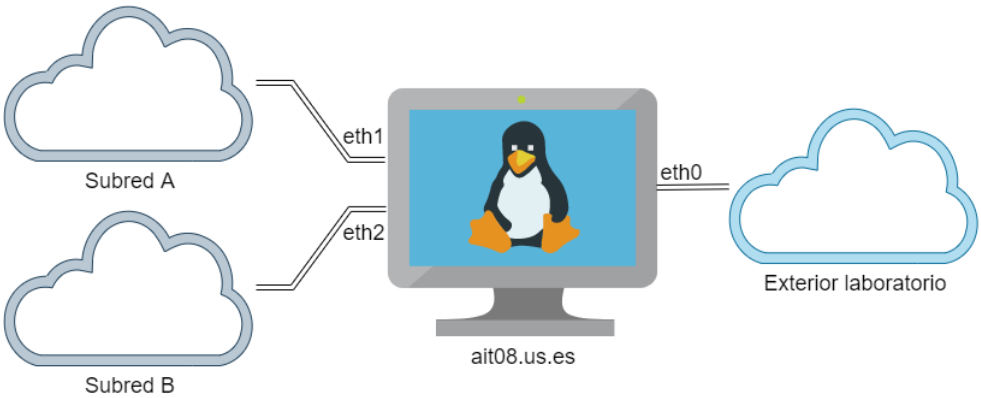

Ilustración 4-1 Esquema de arquitectura de red simplificado.

Algunas algunos de los objetivos que se abordaran son:

- Error al enviar mensaje para realizar Wake on Lan (Arranque en red de equipos), debido a que este se envía siempre por la interfaz asociada a Opengnsys y en el caso del laboratorio siempre se enviará por la interfaz eth0 no llegando nunca al equipo que se quiere encender.
- El estado de los equipos del laboratorio no muestra correctamente su estado en la página web de gestión de Opengnsys debido a la desconexión de los clientes ogAgent en los sistemas operativos.
- El apagado remoto no se realiza debido a la desconexión debido a la desconexión de los clientes

ogAgent en los sistemas operativos.

- Error al realizar la restauración de una imagen mediante multicast, debido a que al iniciar la transferencia la herramienta udp-sender[90] se asocia a la interfaz configurada por defecto para Opengnsys y esto provoca que no se llegue a iniciar la transferencia ya que los paquetes salen por otra interfaz.
- Error al realizar la restauración de una imagen mediante Torrent a un número elevado de equipos, al realizar pruebas más de 10 equipos algunos de ellos dejaban de conectarse y no continuaban con el proceso.
- El arranque local del cliente ogLive desde la cache de local de los equipos no arranca correctamente.
- Se quiere de disponer una forma sencilla de poder realizar diferentes acciones a múltiples de equipo de forma sencilla y visual, las acciones que se quieren realizar es apagar o encender un equipo, enviar archivos o procedimientos durante la ejecución del cliente ogLive o ejecutar de comandos.
- Se quiere incluir de forma integrada en la barra superior de la web de gestión de Opengnsys una serie de enlace y pagina que se usará para acceder a funcionalidades que faciliten la administración del laboratorio de Telemática.
- Se quiere hacer uso de la funcionalidad de Opengnsys la ejecución de tareas automáticas al iniciar el cliente ogLive conectados al servidor con el fin de poder restaurar ficheros puntuales.
- Se quiere dispone de un menú en el cliente ogLive que permita realizar un cierto de acciones pero que sea accedido por un usuario que solo tenga acceso a estas acciones y no a la web de gestión de Opengnsys.

Antes empezar a detallar las diferentes acciones que se han llevado a cabo para abordar los objetivos mencionados, se remitirá la página del manual de Opengnsys "Gestión de las imágenes sincronizadas desde la consola"[91], donde se detalla cómo crear y acceder uso de imagen incrementales.

Este tipo de imágenes permite reducir el espacio consumido si se dispone de imágenes muy parecidas y además si se crearán en modo directorio se podría modificar la imagen directamente en el servidor para introducir cambios sin tener que generar una imagen nueva. Se han realizado pruebas sobre la versión 1.1.1c y no se ha detectado ningún fallo en su funcionamiento.

Tambien se ha de mencionar las modificaciones realizada para poder hacer uso de Opengnsys en el clúster de alta disponibilidad detalladas en la subsección Adaptación de Opengnsys para su uso en el clúster de alta disponibilidad

4.2.2 Actualización de la versión de Opengnsys a la versión 1.1.1c

La primera acción que se ha llevado a cabo es realizar la actualización de la versión de Opengnsys para pasar de la versión 1.0.6 a la versión 1.1.1c.

Tras realizar la actualización se realizaron pruebas y llego a la conclusión que los siguientes objetivos se vieron solventado:

- El estado de los equipos del laboratorio no muestra correctamente su estado en la página web de gestión de Opengnsys debido a la desconexión de los clientes ogAgent en los sistemas operativos.
- El apagado remoto no se realiza debido a la desconexión debido a la desconexión de los clientes ogAgent en los sistemas operativos.
- Error al realizar la restauración de una imagen mediante Torrent a un número elevado de equipos, al realizar pruebas más de 10 equipos algunos de ellos dejaban de conectarse y no continuaban con el proceso.

Para los dos primeros objetivos es necesario la instalación de los clientes de ogClient para dicha versión en los sistemas operativos, esto se detallará en el

ANEXO I: Modificaciones y procedimientos de opengnsys.

Para realizar la actualización, autentificado como superusuario, se debe seguir los siguientes pasos:

1. Descargar los ficheros de instalación mediante el uso de la herramienta wget[92].

```
 wget          https://opengnsys.es/trac/downloads/opengnsys-1.1.1c-
r20200615.828277b.tar.gz
```

El argumento usado es:

https://opengnsys.es/trac/downloads/opengnsys-1.1.1c-r20200615.828277b.tar.gz	Se trata de la URL donde se encuentra los ficheros comprimido para la instalación de la versión 1.1.1c

2. Descomprimir el descargado mediante el uso de la herramienta tar.

```
tar -xvzf opengnsys-1.1.1c-r20200615.828277b.tar.gz
```

Los argumentos usados son:

-x	Indica que se va a realizar la operación de descompresión.
-v	Indica que se realice en modo verboso y de esta manera mostrar lo archivos que han sido descomprimido.
-z	Establece que el formato del fichero a descomprimir es gzip.
-f	Indica que el origen de los datos a descomprimir es un fichero.
opengnsys-1.1.1c-r20200615.828277b.tar.gz	Indica el fichero a descomprimir.

3. Ejecutar el script de actualización ubicado en opengnsys/installer/opengnsys_update.sh. Se deberá ejecutar dos veces, la primera actualizará el script de actualización ubicado en /opt/opengnsys/lib/opengnsys_update.sh y la segunda vez realizará el proceso de actualización desde la versión 1.0.6 a la versión 1.1.1c.

```
 opengnsys/installer/opengnsys_update.sh
```

El resultado de la primera ejecución debe ser el siguiente:

```
 OpenGnsys update begins at Mon 30 Nov 00:58:37 CET 2020
 /tmp/opengnsys_update ~
 Files       /root/opengnsys/installer/opengnsys_update.sh       and
/opt/opengnsys/lib/opengnsys_update.sh differ
 OpenGnsys updater has been overwritten
 Please, run this script again
```

1. Ejecutar un script para la corrección de los valores por defecto de algunas columnas, ya que en la versión 1.1.1 algunas columnas pasaron a tener un valor por defecto y no es aplicado este cambio cuando se realiza la actualización de la base de datos mediante el script opengnsys/admin/Database/ogAdmBD-

1.0.6b-1.1.0a.sql. Este error se dio al actualizar opengnsys y como gestor de base de datos mariaDB. Para ello haremos uso de la herramienta mysql[93].

```
mysql -uUsername -pPassword ogAdmBD < aniadirValoresPorDefecto.sql
```

Los argumentos usados son:

-uUsername	Indica el nombre de usuario con el que se va a establecer la conexión para aplicar el script. Este debe ser cambiado en función al usuario configurado para el acceso a la base de datos desde Opengnsys o para el superusuario del gestor de base de datos.
-pPassword	Indica la contraseña del usuario anteriormente indicado.
ogAdmBD	Indica el nombre de la base datos de Opengnsys y sobre la que se aplicará el script.
aniadirValoresPorDefecto.sql	Indica el nombre del script de SQL que se va a aplicar.

El contenido del fichero aniadirValoresPorDefecto.sql deberá ser:

```
ALTER TABLE imagenes
 MODIFY nombreca varchar(50) NOT NULL DEFAULT '';
ALTER TABLE ordenadores_particiones
 MODIFY  idordenador int(11) NOT NULL DEFAULT '0',
 MODIFY  numdisk smallint NOT NULL DEFAULT '0',
 MODIFY  numpar smallint NOT NULL DEFAULT '0',
 MODIFY  codpar int(8) NOT NULL DEFAULT '0',
 MODIFY  tamano int(11) NOT NULL DEFAULT '0',
 MODIFY  idsistemafichero smallint(11) NOT NULL DEFAULT '0',
 MODIFY  idnombreso smallint(11) NOT NULL DEFAULT '0',
 MODIFY  idimagen int(11) NOT NULL DEFAULT '0',
 MODIFY  idperfilsoft int(11) NOT NULL DEFAULT '0';
ALTER TABLE acciones
 MODIFY tipoaccion smallint(6) NOT NULL DEFAULT '0',
 MODIFY idtipoaccion int(11) NOT NULL DEFAULT '0',
 MODIFY descriaccion varchar(250) NOT NULL DEFAULT '',
 MODIFY idordenador int(11) NOT NULL DEFAULT '0',
 MODIFY ip varchar(50) NOT NULL DEFAULT '',
 MODIFY sesion int(11) NOT NULL DEFAULT '0',
 MODIFY idcomando int(11) NOT NULL DEFAULT '0';
ALTER TABLE parametros
 MODIFY nemonico char(3) NOT NULL DEFAULT '',
 MODIFY nomidentificador varchar(64) NOT NULL DEFAULT '',
 MODIFY nomtabla varchar(64) NOT NULL DEFAULT '';
ALTER TABLE tipohardwares
 MODIFY nemonico char(3) NOT NULL DEFAULT '';
ALTER TABLE perfilessoft
  MODIFY  idcentro int(11) NOT NULL DEFAULT '0';
ALTER TABLE programaciones
  MODIFY sesion int(11) NOT NULL DEFAULT '0';
ALTER TABLE aulas
 MODIFY  modomul tinyint(4) NOT NULL DEFAULT '0',
 MODIFY ipmul varchar(16) NOT NULL DEFAULT '';
 MODIFY pormul int(11) NOT NULL DEFAULT '0';
```

```
ALTER TABLE asistentes
 MODIFY  pagina varchar(256) NOT NULL DEFAULT '',
 MODIFY  gestor varchar(256) NOT NULL DEFAULT '',
 MODIFY  funcion varchar(64) NOT NULL DEFAULT '',
 MODIFY  activo tinyint(1) NOT NULL DEFAULT '0';
ALTER TABLE comandos
 MODIFY pagina varchar(256) NOT NULL DEFAULT '',
 MODIFY gestor varchar(256) NOT NULL DEFAULT '',
 MODIFY funcion varchar(64) NOT NULL DEFAULT '',
 MODIFY activo tinyint(1) NOT NULL DEFAULT '0';
ALTER TABLE entornos
 MODIFY  IPerveradm varchar(50) NOT NULL DEFAULT '',
 MODIFY  portserveradm int(20) NOT NULL DEFAULT 2008,
 MODIFY  protoclonacion varchar(50) NOT NULL DEFAULT '';
ALTER TABLE nombresos
 MODIFY nombreso varchar(250) NOT NULL DEFAULT '';
ALTER TABLE  perfileshard
 MODIFY idcentro int(11) NOT NULL DEFAULT '0';
ALTER TABLE  plataforma
 MODIFY plataforma varchar(250) NOT NULL DEFAULT '';
ALTER TABLE procedimientos_acciones
 MODIFY procedimientoid int(11) NOT NULL DEFAULT '0';
ALTER TABLE repositorios
 MODIFY nombrerepositorio varchar(250) NOT NULL DEFAULT '';
ALTER TABLE sistemasficheros
 MODIFY codpar int(8) NOT NULL DEFAULT '0';
ALTER TABLE tiposos
 MODIFY tiposo varchar(250) NOT NULL DEFAULT '',
 MODIFY idplataforma int(11) NOT NULL DEFAULT '0';
ALTER TABLE tipospar
 MODIFY codpar int(8) NOT NULL DEFAULT 0,
 MODIFY tipopar varchar(250) NOT NULL DEFAULT '',
 MODIFY clonable tinyint(4) NOT NULL DEFAULT '0';
```

4.2.3 Corrección en envió de mensajes para realizar Wake On Lan

Para realizar correctamente él envió de los mensajes de Wake On Lan cuando se dispone de varias subredes se han modificado los ficheros del código fuente ogAdmServer.c y ogAdmServer.h y fichero Makefile usado para la compilación.

En la modificación que sea ha realizado se ha incluido el cálculo como variable la dirección IP del cliente a arrancar para el cálculo la interfaz de salida del mensaje de Wake On Lan.

Para el cálculo de la interfaz de salida se hará uso la ejecución del siguiente comando, que hace uso de la herramienta ip y grep.

```
ip route get IPcliente | grep -oP 'dev \K[\w.]+'
```

Los argumentos usados con el comando ip son:

route get	Establece que se calcule ruta para el destino indicado en función a la tabla de encaminamiento.
IPcliente	Indica la dirección IP destino para calcular la ruta.

Los argumentos usados con el comando ip son:

-o	Indica que solo muestre por salida estándar
-P	Indica la dirección IP destino para calcular la ruta.
'dev \K[\w.]+'	Se trata de una expresión regular mediante la cual se obtiene como resultado eth0 de la siguiente cadena (Cadena resultado de la ejecución de la herramienta ip anterior): 178.16.17.216 via 193.147.162.1 dev eth0 src 193.147.162.168 cache

Las rutas donde se encuentran los ficheros son:

- ogAdmServer.h: opengnsys/admin/Sources/Services/ogAdmAgent/sources/ogAdmServer.h
- ogAdmServer.c: opengnsys/admin/Sources/Services/ogAdmAgent/sources/ogAdmServer.c
- Makefile: opengnsys/admin/Sources/Services/ogAdmAgent/Makefile

Para realizar la corrección se ha introducido lo siguiente cambios:

1. En el fichero ogAdmServer.h se incluido la definición de la función execCommandShellScript, esta función permitirá lanzar un comando como si se realizar desde un Shell y recuperar la salida de este. Se debe añadir el siguiente fragmento al final de este:

```
 std::string execCommandShellScript(char* commando);
```

2. El fichero ogAdmServer.c vuelve a ser códigos C++ y se renombra a ogAdmServer.cpp y se modifica el Makefile usado para su complicación. El contenido del Makefile se debe sustituir por el siguiente:

```
# makefile
# Nombre del proyecto
PROYECTO := ogAdmServer
# Directorio de instalación
INSTALL_DIR := /opt/opengnsys
# Opciones de compilacion
CFLAGS := $(shell mysql_config --cflags)
CFLAGS += -g -Wall -I../../Includes
# Opciones de linkado
LDFLAGS := -Wl,--no-as-needed $(shell mysql_config --libs) -lev -
ljansson -ldbi
# Ficheros objetos
OBJS := sources/ogAdmServer.o sources/dbi.o
all: $(PROYECTO)
$(PROYECTO): $(OBJS)
 g++ $(LDFLAGS) $(CFLAGS) $(OBJS) -o $(PROYECTO)
install: $(PROYECTO)
 cp $(PROYECTO) $(INSTALL_DIR)/sbin
```

```
 cp $(PROYECTO).cfg $(INSTALL_DIR)/etc
clean:
 rm -f $(PROYECTO) $(OBJS)
uninstall: clean
 rm -f /usr/local/sbin/$(PROYECTO) /usr/local/etc/$(PROYECTO).cfg
sources/%.o: sources/%.cpp
 g++ $(CFLAGS) -c -o"$@" "$<"
sources/%.o: sources/%.c
 g++ $(CFLAGS) -c -o"$@" "$<"
```

3. En el fichero ogAdmServer.cpp se añade la implementación de la función execCommandShellScript. Para ello se debe añadir el siguiente fragmento tras la implementación de la función wake_up_broadcast:

```
 //________________________________________________________________
 // Función: execCommandShellScript
 //
 //Descripción:
 //Ejecuta el comando pasado en la cadena command
 //
 //Parámetros de entrada:
 // - command : Cadena con el contenido del command a ejecutar
 //________________________________________________________________

 std::string execCommandShellScript(char* command) {
     char buffer[128];
     std::string result = "";
     FILE* pipe = popen(command, "r");
     if (!pipe) throw std::runtime_error("popen() failed!");
     try {
         while (fgets(buffer, sizeof buffer, pipe) != NULL) {
             result += buffer;
         }
     } catch (...) {
         pclose(pipe);
         throw;
     }
     pclose(pipe);
     //elimino un salto de linea provacado por fgets
     if(result != ""){
       result.erase(result.end()-1);
     }
     return result;
}
```

4. En el fichero ogAdmServer.cpp se modifica la función wake_up_broadcast para que haga uso de la función execCommandShellScript para ejecutar el comando antes mencionado para el cálculo de la interfaz de salida. Para ello se debe sustituir completamente la implementación de la función por la siguiente:

```
static bool wake_up_broadcast(int sd, struct sockaddr_in *client,
      const struct wol_msg *msg, char *ip)
{
```

```
struct sockaddr_in *broadcast_addr;
struct ifaddrs *ifaddr, *ifa;
int ret;
 if (getifaddrs(&ifaddr) < 0) {
  syslog(LOG_ERR, "cannot get list of addresses\n");
  return false;
 }
 client->sin_addr.s_addr = htonl(INADDR_BROADCAST);
 std::string IPtring(ip);
 std::string firstCommand =  "ip route get "+ IPtring + "| grep -oP
'dev \\K[\\w.]+'";
 char firstCommandArray[firstCommand.size() + 1];
 firstCommand.copy(firstCommandArray, firstCommand.size() + 1);
 firstCommandArray[firstCommand.size()] = '\0';
 std::string                        interfaz_out                        =
execCommandShellScript(firstCommandArray);

 if(interfaz_out == ""){
  interfaz_out = std::string(interface);
 }

 for (ifa = ifaddr; ifa != NULL; ifa = ifa->ifa_next) {
  if (ifa->ifa_addr == NULL ||
    ifa->ifa_addr->sa_family != AF_INET ||
    strcmp(ifa->ifa_name, interfaz_out.c_str()) != 0)
    continue;
  broadcast_addr = (struct sockaddr_in *)ifa->ifa_ifu.ifu_broadaddr;
  client->sin_addr.s_addr = broadcast_addr->sin_addr.s_addr;
  break;
 }
 freeifaddrs(ifaddr);
 ret = sendto(sd, msg, sizeof(*msg), 0, (struct sockaddr *)client,
sizeof(*client));
 if (ret < 0) {
  syslog(LOG_ERR, "failed to send broadcast wol\n");
  return false;
 }
 return true;
}
```

5. En el fichero ogAdmServer.cpp se modifica la llamada a la función wake_up_broadcast para que se le pase como argumento la dirección IP del cliente también. Para ello se debe sustituir la siguiente llamada:

```
 ret = wake_up_broadcast(s, &WakeUpCliente, &Trama_WakeUp);
```

Por la esta llamada:

```
 ret = wake_up_broadcast(s, &WakeUpCliente, &Trama_WakeUp, iph);
```

Las modificaciones indica se deben realizar sobre el código fuente ante realizar una actualización o instalación de Opengnsys, ya que durante estos procesos se realizará la compilación del binario con la modificación incluida.

De no hacer sería necesario compilarlo mediante los archivos proporcionado en la memoria donde se incluye en un directorio la dependencia necesaria, los ficheros con las modificaciones realizada y un nuevo Makefile para

realizar la compilación y sustitución del antiguo binario.

4.2.4 Corrección al realizar la restauración mediante multicast debido a las múltiples subredes del laboratorio

La corrección realizada permite hacer uso la restauración Multicast en un escenario en el que el servidor donde se aloja el servidor de opengnsys es la puerta de acceso a los clientes de Opengnsys y se posee varias subredes.

Esto es necesario debido a que udp-sender se asocia directamente a la interfaz con la que se inicia la comunicación no permitiendo usar tablas de enrutamiento o método similares para reenviar el tráfico por una interfaz concreta.

Esta modificación no es válida en el caso en que el servidor de Opengnsys no sea la puerta de enlace debido a este comando que es usado para determinar la dirección IP que se usara para establecer la comunicación entre cliente y servidor.

```
ip route show 0.0.0.0/0 | cut -d\ -f3
```

Sería necesario encontrar una alternativa donde en la que tanto el cliente como el servidor puede determinar la dirección IP de la interfaz del servidor con la cual se comunica ambos.

Para realizar la corrección se ha modificado los siguientes ficheros:

Fichero	Ubicación tras instalación	Ubicación antes de la instalación
restoreImage	/opt/opengnsys/client/scripts/restoreImage	opengnsys/tclient/shared/scripts/restoreImage
ogAdmRepoAux	/opt/opengnsys/sbin/ogAdmRepoAux	opengnsys/admin/Sources/Services/ogAdmRepoAux
getRepoIface	/opt/opengnsys/bin/getRepoIface	opengnsys/repoman/bin/getRepoIface
sendFileMcast	/opt/opengnsys/bin/sendFileMcast	opengnsys/repoman/bin/sendFileMcast
Protocol.lib	/opt/opengnsys/client/lib/engine/bin/Protocol.lib	opengnsys/client/engine/Protocol.lib

Para realizar la corrección se ha introducido lo siguiente cambios:

1. En el script restoreImage se una espera de un tiempo aleatorio antes de realizar la llamada ogMcastReceiverPartition y tras la llamada ogMcastRequest, esto permite que en el servidor se haya iniciado correctamente la sesión multicast antes de que se solicite conexión desde el cliente. Si no se hiciera esta espera y no se hubiera iniciado la sesión cuando el cliente solicita la conexión se perdería la petición y tanto en el cliente como en el servidor saltaría el temporizador de espera y no se realizaría la restauración correctamente. Para ello se debe sustituir el siguiente fragmento de código:

```
  ogMcastRequest "$IMGNAME.img" "$PROTOOPT" || exit $?
  ogExecAndLog command ogMcastReceiverPartition "$DISK" "$PART" "$PORT"
"$TOOL" "$COMPRESS"
```

Por este fragmento de código:

```
ogMcastRequest "$IMGNAME.img" "$PROTOOPT" || exit $?
ogExecAndLog command ogMcastReceiverPartition "$DISK" "$PART"
NUMBER=$[ ( $RANDOM % 30 )  + 10 ]
```

```
  ogEcho log session "[40] Esperando $NUMBER segundos para iniciar la
conexion multicast"
  sleep $NUMBER
  ogExecAndLog command ogMcastReceiverPartition "$DISK" "$PART" "$PORT"
"$TOOL" "$COMPRESS"
```

2. En el script ogAdmRepoAux se modifica para realizar la lectura de un cuarto argumentos cuando se solicita iniciar la sesión para realizar un envió mediante multicast y pasarlo como argumento del script sendFileMcast. Este parámetro será la dirección IP a la cual debe asociar la sesión multicast que se va a crear. Para ello se debe sustituir el siguiente fragmento:

```
   echolog "Ejecutar $(which sendFileMcast) $FILE $MCASTOPT"
   sendFileMcast $FILE $MCASTOPT &>> $MCASTLOG
```

Por este fragmento:

```
  IPINTEFAZ="$PARM4"
  echolog "Ejecutar $(which sendFileMcast) $FILE $MCASTOPT $IPINTEFAZ"
  sendFileMcast $FILE $MCASTOPT $IPINTEFAZ &>> $MCASTLOG
```

3. En el script getRepoIface se modifica para recibir un posible argumento que será la dirección IP para la cual debe calcular la interfaz en la que está asociada, de no recibirlo calcula la interfaz en función al valor de la variable IPlocal del fichero de configuración ogAdmRepo.cfg. Para ello se debe añadir el siguiente fragmento tras la importación de las variables del fichero de configuración mediante el uso de la herramienta source[94]:

```
  if [ $# -eq 1 ]; then
      IPlocal="$1"
  fi
```

4. En el script sendFileMcast se modifica para permitir recibir un tercer parámetro opcional, este representa una la dirección IP a la cual debe asociar la sesión multicast que se va a crear. Si no se recibiera este parámetro o no la dirección IP no está asociada a ninguna interfaz de la maquina se hará aquella configurada en el fichero de configuración ogAdmRepo.cfg Para ello se debe sustituir el siguiente fragmento de código:

```
  REPO_IFACE="$(/opt/opengnsys/bin/getRepoIface)"

  # Si se solicita, mostrar ayuda.
  if [ "$*" == "help" ]; then
      echo "Formato: $PROG fichero|nombreImagen datosMulticast"
      echo "Ejemplo: $PROG /opt/opengnsys/images/imagen1.pgz 9000:full-
duplex:239.194.17.2:70M:20:120"
      echo "Ejemplo: $PROG imagen1 9000:full:239.194.17.2:70M:20:120"
      exit 0
  fi
  # Error si no se reciben 2 parámetros.
  if [ $# -ne 2 ]; then
      echo   "$PROG   Error:   Formato:   $PROG   fichero|nombreImagen
datosMulticast"
      exit 1
```

```
  fi
```

Por este fragmento de código:

```
  # Si se solicita, mostrar ayuda.
  if [ "$*" == "help" ]; then
      echo   "Formato:   $PROG   fichero|nombreImagen   datosMulticast
[IPaAsociar]"
      echo "Ejemplo: $PROG /opt/opengnsys/images/imagen1.pgz 9000:full-
duplex:239.194.17.2:70M:20:120"
      echo "Ejemplo: $PROG imagen1 9000:full:239.194.17.2:70M:20:120"
      echo "Ejemplo: $PROG /opt/opengnsys/images/imagen1.pgz 9000:full-
duplex:239.194.17.2:70M:20:120 172.17.2.1"
      echo  "Ejemplo:  $PROG  imagen1  9000:full:239.194.17.2:70M:20:120
172.17.2.1"
      exit 0
  fi

  # Error si no se reciben 2 parámetros.
  if [ $# -le 2 ]; then
      echo   "$PROG   Error:   Formato:   $PROG   fichero|nombreImagen
datosMulticast [IPaAsociar]"
      exit 1
  fi

  if [ $# -eq 3 ]; then
      REPO_IFACE="$(/opt/opengnsys/bin/getRepoIface $3)"
      if [ "$REPO_IFACE" == "" ]; then
         REPO_IFACE="$(/opt/opengnsys/bin/getRepoIface)"
      fi
  else
      REPO_IFACE="$(/opt/opengnsys/bin/getRepoIface)"
  fi
```

5. En el fichero Protocol.lib se modifica la función ogMcastRequest para en la petición de inicio de la sesión multicast se envía un cuarto argumento que será la dirección IP con la cual se realizará iniciará la sesión en el servidor, esta será la asociada a la interfaz que pertenece a la misma subred donde se encuentra el equipo o los equipos a restaurar y además se cambia la dirección IP para la cual se debe comprobar que se encuentra abierto el puerto de la sesión multicast a iniciar. Para ello se debe sustituir el contenido de la función por el siguiente:

```
  function ogMcastRequest ()
  {
  # Variables locales
  local FILE PROTOOPT PORT PORTAUX REPOIP REPOPORTAUX REPEAT OGUNIT

  # Si se solicita, mostrar ayuda.
  if [ "$*" == "help" ]; then
      ogHelp "$FUNCNAME" "$FUNCNAME path_filename str_mcastoptions"
      return
  fi
  # Error si no se reciben 2 parámetros.
  [ "$#" == 2 ] || ogRaiseError $OG_ERR_FORMAT || return $?

  OGUNIT="$(df|awk '/ogimages/ {print $1}'|cut -d/ -f5)/"
```

```
  FILE="$OGUNIT$1"
  PROTOOPT="$2"

  #TODO: CONTROL PARAMETROS

  PORT=$(echo $2 | cut -f1 -d":")
  let PORTAUX=$PORT+1
  REPOIP=$(ogGetRepoIp)
  REPOIPAUX=$(ip route show 0.0.0.0/0 | cut -d\  -f3)
  REPOPORTAUX=2009
  REPEAT=0
  until nmap -n -sU -p $PORTAUX $REPOIPAUX | grep open
  do
      let REPEAT=$REPEAT+1
      [   "$REPEAT"   -lt   6   ]   ||   ogRaiseError   session   log
  $OG_ERR_PROTOCOLJOINMASTER   "MULTICAST   \"$FILE\"   \"$PROTOOPT\"
  $FILELIST" || return $?
      echo "$MSG_SCRIPTS_TASK_START :  hose $REPOIP $REPOPORTAUX --out
  sh -c "echo -ne START_MULTICAST $FILE $2""
      #update-cache:
      hose $REPOIP $REPOPORTAUX --out sh -c "echo -ne START_MULTICAST
  "$FILE" "$PROTOOPT" "$REPOIPAUX""
      #multicas-direct:   hose  $REPOIP  2009  --out  sh  -c  "echo  -ne
  START_MULTICAST /$IMAGE.img $OPTPROTOCOLO"
      sleep 10
  done
  }
```

6. En el fichero Protocol.lib se modifica la función ogMcastSyntax para que al iniciar la descarga mediante multicast siempre añada como dirección IP desde la que recibirá la información se la misma con la que se solicitó el inicio de la sesión multicast en la función ogMcastRequest. Para ello se cambia el contenido tras el siguiente comentario hasta el cierre de la cláusula else:

```
 # Deteccion automatica de la subred del cliente para anadir la  IP del
repositorio a la orden udp-receiver en el caso de encontrarse en distinta
subred del repo
```

Por este fragmento de código:

```
 REPOIP="$(ip route show 0.0.0.0/0 | cut -d\  -f3)"
 SERVERADDRESS=" --mcast-rdv-address $REPOIP"
```

Para finalizar se detallará algunas recomendaciones de configuración de las aulas para realizar la transferencia mediante multicast más robusta.

- En la configuración de velocidad máxima de transferencia poner un valor un 20% menor la velocidad mínima de la red. Esto se debe a que si se pone un valor mayor la red se verá colapsa en el proceso de transferencia.
- Configurar dirección IP de multicast y puerto para la transferencia multicast distinto para cada subred. Esto se debe a que de no hacerlo no sería posible realizar transferencia simultaneas en subredes distintas debido a conflicto de puerto en uso y establecimiento de conexión a direcciones IP asociada para la transferencia en otra subred.

4.2.5 Corrección de arranque del cliente ogLive desde cache y configuración del menú de este.

Se han realizado prueba sobre el procedimiento indicado en la página del manual de Opengnsys "Cliente OpenGnSys iniciado desde cache"[95] y tras esta se observaron dos fallos diferentes:

- Se sigue produciendo el error mencionado en el foro de Opengnsys[96] con el número de referencia #351. En las pruebas se producía al crear la partición cache para su posterior uso con cliente de ogLive con la versión de kernel anterior a 4.15.

  Para solucionar este error al crear la cache se deben seguir lo siguiente pasos con el equipo iniciado el cliente de ogLive.

  1. Desmontar la partición donde se encuentra ubicada la cache mediante el uso de la herramienta umount.

```
umount /dev/particionCache
```

El argumento usado es:

/dev/particionCache	Indica la partición que se va a desmontar.

  2. Comprobar y reparar los posibles fallos que pudiera existir en la partición donde se encuentra ubicada la cache mediante el uso de la herramienta e2fsck[97].

```
e2fsck -y /dev/particionCache
```

Los argumentos usados son:

-y	Indica que siempre se intente la reparación de los fallos que existieran en la partición.
/dev/particionCache	Indica la partición sobre la que se realizará la operación.

  3. Deshabilitamos las características del sistema de fichero metada_crc y 64bit la partición donde se encuentra ubicada la cache y de esta manera permitir su uso con kernel inferior al 4.15, Para ello se hará uso de la herramienta tune2fs[98].

```
tune2fs -O ^metadata_crc,^64bit  /dev/particionCache
```

Los argumentos usados son:

-O	Indica que se va a habilitar o deshabilitar una de las características del sistema de fichero.
^metadata_crc, ^64bit	Establece las características las cuales se va a deshabilitar.
/dev/particionCache	Indica la partición sobre la que se realizará la operación.

  4. Volver a comprobar y reparar los posibles fallos que pudiera existir en la partición donde se encuentra ubicada la cache mediante el uso de la herramienta e2fsck.

```
 e2fsck -y -f /dev/particionCache
```

Los argumentos usados son:

-y	Indica que siempre se intente la reparación de los fallos que existieran en la partición.
-f	Fuerza la comprobación, aunque la partición parezca no contener fallos
/dev/particionCache	Indica la partición sobre la que se realizará la operación.

5. Liberar los bloques de los cuales no se hará uso al deshabilitar la característica 64bit del sistema de fichero. Para ello se hará uso de la herramienta resize2fs[99].

```
resize2fs -s /dev/particionCache
```

Los argumentos usados son:

-s	Establece que se deshabilite la característica 64bit y libere aquellos bloques que no serán usado al establecer el límite en 2^32 bloques.
/dev/particionCache	Indica la partición sobre la que se realizará la operación.

- Al hacer uso de imágenes de ogLive bionic 5.0.0 y posteriores hemos detectado que se producía un error durante la cargar del menú del cliente, realizando un investigación de la causa se llegó a la conclusión de que se debe a que no se monta correctamente la cache en el inicio y esto se debe a que en las estas versiones se han eliminado en el archivo ogclient.sqfs los directorios cache, images y log que se encontraba en la ruta /opt/opengnsys/.

  Para corregir las imágenes sea seguir los siguientes pasos estando autentificado como superusuario:

  1. Descargar las herramientas ogLive-Builder mediante desde su repositorio mediante el uso de git.

```
git clone https://github.com/opengnsys/ogLive-Builder.git
```

Los argumentos usados son:

clone	Indica que se va a clonar un repositorio remoto.
https://github.com/opengnsys/ogLive-Builder.git	Indica la URL del repositorio de git a descargar.

2. Seleccionar la imagen que se va a modificar, para ello es necesario seleccionar esa imagen como la imagen por defecto para Opengnsys. Para ello se hace uso del script oglivecli.

```
/opt/opengnsys/bin/oglivecli set-default indice
```

Los argumentos usados son:

set-default	Indica que se va a cambiar la imagen por defecto que se usara a la hora de iniciar Opengnsys en los equipos clientes.
indice	Indica el valor del índice de la imagen disponible que se va a seleccionar como por defecto. Se puede consultar el índice asociado a cada imagen mediante usando como argumento list del script oglivecli.

3. Crear el fichero img a partir del sqfs para poder montarlo y de esta manera modificarlo para realizar

la corrección. Se hará uso del script ogclientSqfs2Img.

```
ogLive-Builder/ogclientSqfs2Img
```

4. Montar el fichero img para realizar las modificaciones. Se hará uso de la herramienta mount.

```
mount                    /opt/opengnsys/tftpboot/ogclient/ogclient.img
/opt/opengnsys/tftpboot/ogclient/ogclientmount -o loop,offset=32256
```

Los argumentos usados son:

/opt/opengnsys/tftpboot/ogclient/ogclient.img	Indica la imagen a montar.
/opt/opengnsys/tftpboot/ogclient/ogclientmount	Indica el directorio donde se montará la imagen.
-o loop,offset=32256	Establece las opciones de montajes loop, que permite montar la imagen y modificarla al modificar el directorio, y offset, indica el punto desde inicio de la imagen que será usado como inicio al montar la imagen.

5. Crear los directorios faltantes cache, images y log. Se hará uso de la herramienta mkdir.

```
mkdir /opt/opengnsys/tftpboot/ogclient/ogclientmount/opt/opengnsys/log
mkdir
/opt/opengnsys/tftpboot/ogclient/ogclientmount/opt/opengnsys/images
mkdir
/opt/opengnsys/tftpboot/ogclient/ogclientmount/opt/opengnsys/cache
```

El argumento usado es:

/opt/opengnsys/tftpboot/ogclient/ogclientmount/opt/opengnsys/X	Estable el directorio a crear.

6. Generar el fichero sqfs a partir del img modificado. Se hará uso del script ogclientImg2Sqfs.

```
ogLive-Builder/ogclientImg2Sqfs
```

Para finalizar se detallará algunos datos sobre la instalación del arranque local.

- Tras la ejecución del installOfflineMode en cliente ogLive con el que se instala el cliente en la cache local, se copiará en la ruta del cliente /opt/opengnsys/cache/menus/ un fichero HTML cuyo nombre será la dirección IP asociada al cliente al arrancar ogLive. Este menú se genera automáticamente para mostrar los sistemas operativo disponible en el equipo e iniciarlo. Este puede ser personalizado siguiendo el manual de Opengnsys "Creación de menús de inicio personalizados para los ordenadores clientes"[100] para incluir lanzado de script que permitan restaurar imagen o realizar otras acciones. Un ejemplo de cómo añadir una opción para permitir restaurar una imagen desde cache seria:

```
<p><a href="commandwithconfirmation:/opt/opengnsys/scripts/restoreImage
```

```
CACHE ubuntu 1 1">Restaurar ubuntu desde cache</a></p>
```

- En la página del manual de Opengnsys "Cliente OpenGnSys iniciado desde cache" se indica que para iniciar el cliente desde grub es necesario ejecuta el ogGrubAddOgclient pero desde la versión 1.1.0 fue renombrado a ogGrubAddOgLive.

4.2.6 Uso de la propiedad de configuración autoexec para realizar la restauración de fichero en el arranque

La propiedad de configuración Autoexec se trata de la asociación de un procedimiento que puede ser configurado a nivel de equipo, una vez configurado este procedimiento es ejecutado cada vez que se inicia el cliente ogLive antes de realizar otra acción.

Esta característica de Opengnsys se usará en el laboratorio para realizar la restauración de ficheros puntuales, para ello se creará un procedimiento principal que agruparan diferentes procedimientos que se encargaran de realizar la copia de los diferentes ficheros. Esto fue una duda preguntada en el foro de Opengnsys y su número de referencia #236

Los procedimientos que realizarán las copias serán creados siendo la ejecución de un script que contendrá como contenido:

```
 ogEcho log session "Copiando ruta_origen en el disco num_disco
num_particion en ruta_destino"
 ogCopyFile REPO ruta_origen num_disco num_particion ruta_destino
 ogEcho log session "Copiado ruta_origen en el disco num_disco
num_particion en ruta_destino"
```

Con este contenido se realizará las siguientes acciones:

- Logear y mostrar en la pantalla del cliente cuando se inicia la copia y cuando se termina mediante el uso de la función ogEcho y cuyos argumentos son:

log	Indica que se escriba en el fichero de logs el mensaje indicado.
session	Indica que se muestre por pantalla el mensaje indicado.
Mensaje	Indica el mensaje que se quiere mostrar y logear.

- Realizar la copia de los ficheros deseado en una ruta determinada en una disco y partición concreta mediante el uso de la función ogCopyFile y cuyos argumentos son:

REPO	Indica que el origen de los ficheros será la ruta absoluta /opt/opengnsys/images/ que es montada desde el servidor mediante samba iniciar el cliente ogLive.
ruta_origen	Indica la ruta relativa de los ficheros o directorios origen a copiar partiendo de /opt/opengnsys/images/. En caso de indicar un directorio y este no terminar con / se copiará creará una carpeta con el mismo nombre en el destino en vez de copiar el contenido de este.
num_disco	Indica número del disco destino de la copia.
num_particion	Indica número de la partición destino de la copia.

ruta_destino	Indica la ruta destino a realizar la copia en el disco y partición seleccionada. Esta ruta debe ser una ruta absoluta para el disco y partición.

Un ejemplo de procedimiento para realizar la restauración del fichero passwd sería el siguiente:

```
  ogEcho log session "Copiando centos/passwd en el disco 1 2 en
/etc/passwd"
  ogCopyFile REPO centos/passwd 1 2 /etc/passwd
  ogEcho log session "Copiado centos/passwd en el disco 1 2 en /etc/passwd
"
```

4.2.7 Modificación para crear menú para operadores

Se quiere disponer de menú alternativo que se accesible desde el cliente ogLive que permita realizar operaciones menos privilegios que tendrían un administrador, el motivo de esto es que se quiere que desde este menú se pueda realizar acciones como puede restaurar una partición, pero no actualizar una imagen.

Para poder disponer de esta función se hará uso de dos elementos nuevos:

- Un nuevo tipo de usuario operador, este solo tendrá acceso desde el cliente ogLive y al autentificarse con este tipo mostrará la parte del menú formado por los elementos de operador.
- Un nuevo tipo elemento de operador para los menús que se compone a partir de los procedimientos definidos, este solo se mostrará cuando se autentifique un usuario de tipo operador.

A continuación, se indicará los diferentes archivos que se han modificados y el porqué de estas modificaciones, partiremos de que la ruta inicial es /opt/opengnsys/www si se realicen la modificación tras la instalación de opengnsys o opengnsys /admin/WebConsole/ en caso de que se realicen sobre el código fuente de Opengnsys.

- El archivo includes/constantes.php se modifica para añadir el código de los tipos de elementos de los menús de clientes el tipo operador.
- El archivo principal/administracion.php se modifica la función SubarbolXML_superadministradores para mostrar menú lateral a los usuarios de tipo operador y la función CreacontextualXMLUsuarios para poder crear usuarios de tipo operador. En caso de realizar la instalación también habría que modificar principal/administracion.device.php.

- El archivo varios/acceso_operador.php se modifica el control de la variable de sesión swop que controlaba si se había autentificado un usuario correctamente para recuperar los elementos privados del menú, ahora se hará lo mismo, pero con la variable swoptipo que además almacenara si el usuario es un administrador (con el valor a 1) o un operador (con el valor a 2). De no esta instanciada la variable swoptipo se mostrará los elementos públicos, si su valor es 1 se mostrará los elementos privados y si es 2 se mostrará los elementos de operador.
- El archivo varios/accesoperadores.php se modifica para que en función del usuario que autentifique se envíe indicando que se muestre los elementos privados o de operador.
- El archivo varios/accionmenu.php se modifica la función pintaMenus para mostrar como otra posible tipo operador en el desplegable para indica el tipo de elemento de la acción del menú.
- El archivo varios/informacion_menus.php se modifica la función SubarbolXML_Items para mostrar las acciones de tipo operador que posee un menú concreto.
- El archivo varios/menucliente.php se modifica para imprimir en el cliente el menú con los elementos

de operador cuando se autentifique con un usuario de tipo operador.

- El archivo controlpostacceso.php se modifica las consultas en la función toma_datos para permitir recuperar usuarios de tipo operador y de esta manera autentificar correctamente con este tipo de usuario, pero se limita se limita para que su acceso únicamente sea posible si se realiza desde un cliente ogLive.
- El archivo validacion/functions.php se modifica la función RecuperaMenu para imprimir un texto que indique que se ha autentificado como operador o como administrador y usar el número de columnas configurado para los elementos privados, esto se debe a que si no sería necesario modificar la base de datos para añadir un campo más.
- Por último, en los siguientes ficheros se ha modificado las traducciones para añadir los elementos necesarios para mostrar las traducciones de los elementos relacionado con el menú de operador, elementos de tipo operador y usuario de tipo operador donde x representa las cadenas esp, cat y eng que se asocia a las traducciones en español, catalán e inglés respectivamente.

```
idiomas/php/x/accionmenu_x.php
idiomas/php/x/administracion_x.php
idiomas/php/x/informacion_menus_x.php
idiomas/php/x/menucliente_x.php
```

Las modificaciones realizadas sobre los distintos ficheros se detallarán en el

Tras realizar las modificaciones se puede observar los siguientes cambios el web de gestión:

- Es posible crear y gestionar usuarios de tipo operador desde el menú de superadministrador destinado a la gestión de usuarios.

Ilustración 4-2 Menú de creación de usuario de tipo operador

Ilustración 4-3 Unidad administrativa en la cual se encuentra un operador asignado

- Es posible de la gestión de elementos de tipo operador en los menús creado a partir de procedimientos.

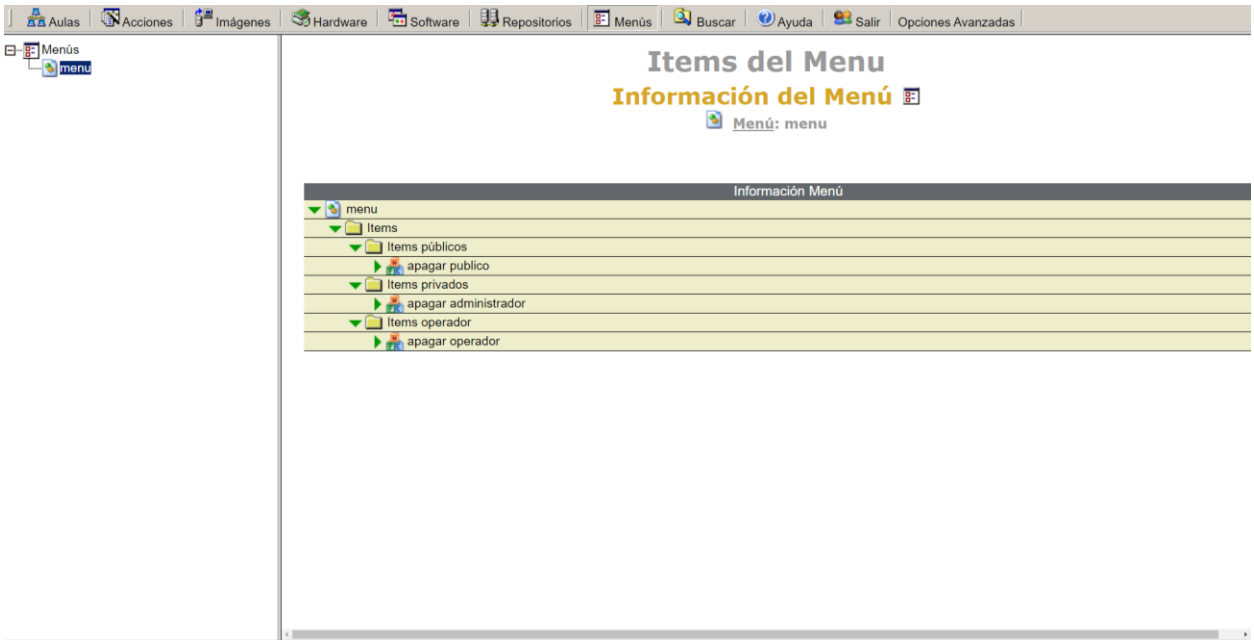

Ilustración 4-4 Información de menú que contiene los diferentes tipos de elementos

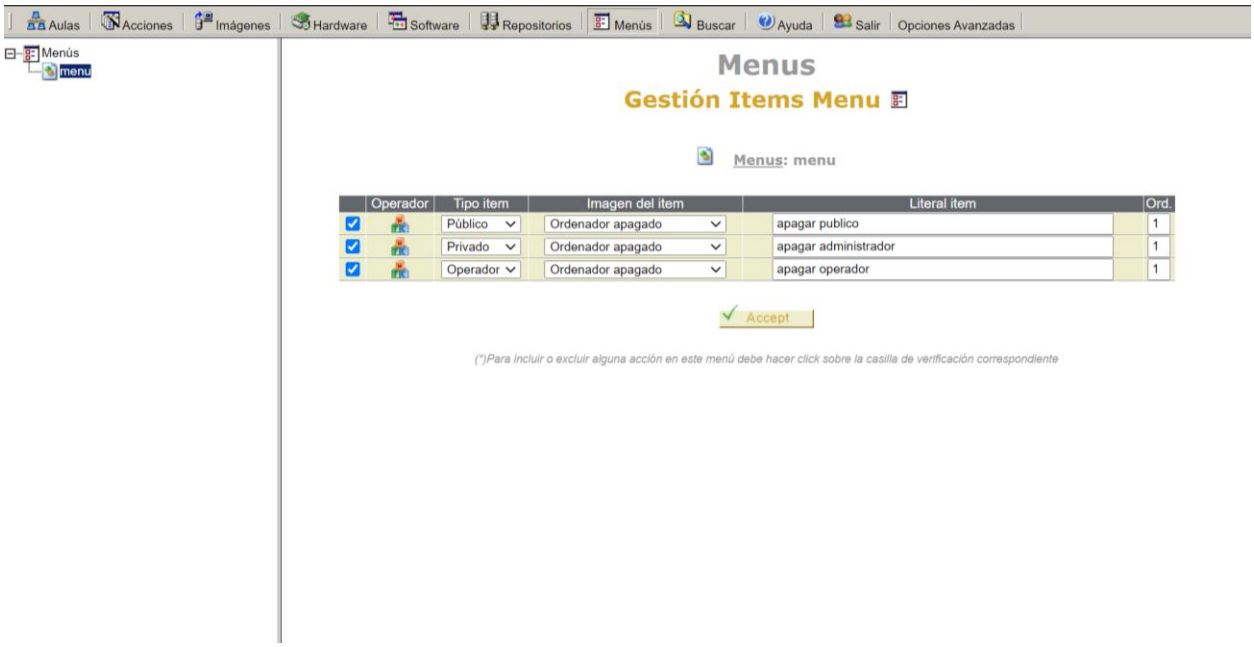

Ilustración 4-5 Gestión de un menú con los diferentes tipos de elementos

- En el cliente ogLive al autentificarnos con un usuario de tipo administrador u operador mostrará información distinta.

Ilustración 4-6 Menú para usuario de tipo administrador

Ilustración 4-7 Menú para usuario de tipo operador

4.2.8 Modificación de la barra superior de la web de gestión de Opengnsys

Mediante esta modificación se quiere tener la posibilidad de integrar dentro de web de gestión de Opengnsys una serie de acceso y páginas de utilidad que faciliten la administración del laboratorio de Telemática.

Entre los elementos que se incluirá:

- Enlace a la web de cliente de Apache Guacamole[101], este se abrirá en una nueva pestaña del navegador.
- Página de sincronización de cambios de Opengnsys en Apache Guacamole, esta página permitirá crear o actualizar las conexiones de acceso remoto en Guacamole de para los equipos configurado en Opengnsys.
- Gestor de fichero basado en el proyecto de GitHub jcampbell1/simple-file-manager[102], este permitirá administrar los ficheros que será usado posteriormente en la siguiente subsección para ser enviado a los clientes ogLive. Este se abrirá en una nueva pestaña del navegador.
- Pagina con información útil acerca del uso de script para la administración de diferentes modelos de switch existen en el laboratorio.

A continuación, se indicará los diferentes archivos que se han modificados o se han creado y el porqué de estas modificaciones, partiremos de que la ruta inicial es /opt/opengnsys/www si se realicen la modificación tras la instalación de opengnsys o opengnsys /admin/WebConsole/ en caso de que se realicen sobre el código fuente de Opengnsys.

- El archivo barramenu.php se para añadir en las funciones definida en el archivo personalizado/modificador_menu.php y la posterior llamada añadirOpcionGestionAvanzada con la cual se creará la opción de opciones avanzada y el elemento con el cual se podrá recorrer los diferentes enlaces o paginas configuradas que se hayan configurado.
- Se modifican los ficheros de idiomas/php/xxx/barramenu_xxx.php para añadir las traducciones de los diferentes enlaces y la opción de opciones avanzada. El valor de xxx variará entre esp, eng y cat dependiendo del idioma.
- Se crea el archivo personalizado/modificador_menu.php donde se definirán los diferentes enlaces y paginas a mostrar en opciones avanzada y contiene las siguientes funciones:
    - añadirOpcionGestionAvanzada: Esta recibe como parámetro la variable que contiene las traducciones de las diferentes opciones y en ella se definen opciones a mostrar con el siguiente formato:

```
 <TD      class="gestion_avanzada"      onmouseout=desresaltar(this)
onmouseover=resaltar(this) align=middle style="display:none;  white-
space: nowrap;">
 <a                  style="text-decoration:                    none"
href="./personalizado/sincronizacion_guacamole.php"
target="frame_contenidos"><span>'.$TbMsg[19].'</a></span>
 </TD>
```

De este formato los parámetros a modificar para crear una nueva opción son:

href	Indica que ruta o localización se carga al pulsar la opción. De incicar una ruta relativa se debe tener en cuenta que el valor inicial de la URL será https://IPervidorOpengnsys/opengnsys/
target	Establece si la carga de la página se realizará dentro de un frame de la web de gestión de Opengys o en otra pestaña. Los posibles valores que se han contemplado son _blank si se desea abrir en una nueva pestaña o frame_contenidos si se desea cargar dentro de la web de Opengnsys.
$TbMsg[19]	Es usada para indicar el texto introducir en la opción. Sería necesario crear una nueva entra en el fichero /php/xxx/barramenu_xxx.php para los distintos idiomas.

  - añadirDespliegueOpciones: Esta recibe el nombre de una clase que identifica a las diferentes opciones y se encargará de añadir la lógica necesaria botón Opciones Avanzadas para mostrar u ocultar las opciones.
  - añadirDesplazamientoIzquierda: Esta recibe el nombre de una clase que identifica a las diferentes opciones y se encargará de añadir la lógica necesaria para mostrar la opción de la izquierda y ocultar la actual.
  - añadirDesplazamientoDerecha: Esta recibe el nombre de una clase que identifica a las diferentes opciones y se encargará de añadir la lógica necesaria para mostrar la opción de la derecha y ocultar la actual.

- Se crea el archivo personalizado/gestor_fichero.php que se trata del proyecto jcampbell1/simple-file-manager al cual se le han realizado las siguientes modificaciones:
  - Se han traducido los textos al español
  - Se eliminado la posibilidad de hacer click sobre un archivo, ya que mediante la configuración de Apache se va a limitar la posibilidad de acceso a los ficheros desde el exterior de la máquina.
  - Se cambiado los iconos y los estilos.
  - Se hace uso del fichero de configuración personalizado/configuracion/configuracion_fichero.php para indicar la ruta de inicio del gestor y limitar el acceso a directorios superiores.
  - Se restringido su acceso para que sea necesario estar autentificado en la web de gestión de Opengnsys para ello se ha incluido el siguiente fragmento al inicio de este:

```
include_once("../includes/ctrlacc.php");
```

En el

ANEXO I: Modificaciones y procedimientos de opengnsys se detallará que configuraciones son necesarias para el uso del gestor de fichero.

- Se incluye la librería jquery en la ruta personalizado/jquery.min.js necesaria para el funcionamiento del gestor de fichero y de esta manera no depender de que sea descargado en cada acceso de la ruta https://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js
- Se crea el archivo personalizado/pagina_gestion_switch.php que se trata de una página escrita para detallar como hacer uso de script de gestión de switch y a la cual se ha limitado el acceso de la misma manera que al gestor de fichero.

- Se crea el archivo personalizado/sincronizacion_guacamole.php que usará para realizar la sincronización de los equipos configurado en Opengnsys y aquellos configurado en Guacamole y realizar las siguientes acciones:
    - Actualizar la dirección IP de aquellos equipos que han cambiado en Opengnsys.
    - Crear conexiones en Guacamoles para nuevos equipos dado de alta en Opengnsys.
    - Eliminar conexiones de Guacamole que no tenga correspondencia con ningún equipo de Opengnsys.
    - Actualización de los parámetros de configuración de las conexiones de Guacamole en caso de que haya cambiado los valores en el archivo personalizado/configuracion/configuracion_sincronizacion_guacamole.php.

  En este archivo se hará uso de consulta a la base datos de Opengnsys para consultar los datos de los equipos configurado en la tabla ordenadores.

  Tambien se seguirá las indicaciones aportada en la página de manual de Guacamole "Modifying data manually"[103] para generar las sentencias SQL para crear las conexiones de los equipos en función a lo indicado en el archivo de configuración personalizado/configuracion/configuracion_sincronizacion_guacamole.php.

  Las conexiones creadas en Guacamole se asignarán a un usuario con todos los permisos disponible y para cada equipo se crearán en un grupo con el nombre del equipo en el que se encontrará las diferentes conexiones con los protocolos configurada con la siguiente estructura:

```
nombreEquipoSubfijoProtocolo
```

  Donde SubfijoProtocolo dependerá del valor configurado en el archivo de configuración para cada protocolo.

En el

ANEXO I: Modificaciones y procedimientos de opengnsys se detallará que el contenido de fichero personalizado/configuración/configuracion_sincronizacion_guacamole.php.

- Se crea el fichero personalizado/sincronizacion_guacamole.js que contendrá las funciones necesarias para realizar la selección múltiple en el archivo personalizado/sincronizacion_guacamole.php y genera un pop-up de confirmación antes de realizar las acciones seccionadas sobre los equipos.
- Se crean los ficheros de idiomas/php/xxx/sincronizacion_guacamole_xxx.php que establecerán las traducciones necesarias para el fichero personalizado/guacamole.php. El valor de xxx variará entre esp, eng y cat dependiendo del idioma.

Se crean los ficheros de configuración personalizado/configuracion/configuracion_sincronizacion_guacamole.php y personalizado/configuracion/configuracion_fichero.php cuyo contenido se detallará en el

ANEXO I: Modificaciones y procedimientos de opengnsys.

Las modificaciones realizadas sobre los distintos ficheros se detallarán en el

ANEXO I: Modificaciones y procedimientos de opengnsys. Y los ficheros nuevos será entregado junto con la memoria.

Tras realizar las modificaciones se puede observar los siguientes cambios el web de gestión:

- Se puede ver y desplegar las opciones avanzadas.

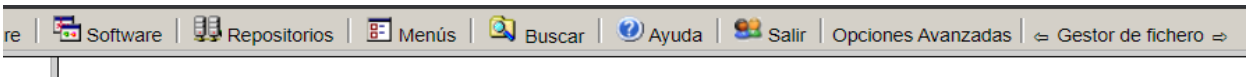

Ilustración 4-8 Opciones Avanzadas desplegada

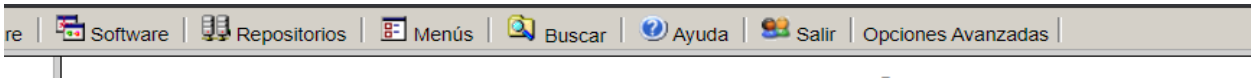

Ilustración 4-9 Opciones Avanzadas replegada

- Se pueden acceder la página de sincronización entre Opengnsys y Guacamole.

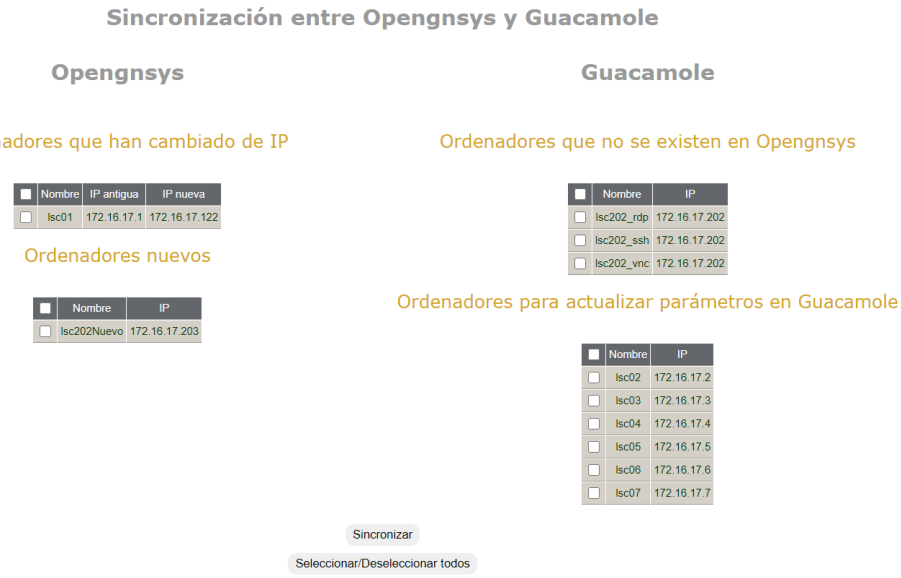

Ilustración 4-10 Página de sincronización entre Opengnsys y Guacamole

- Se podrá acceder al gestor de ficheros.

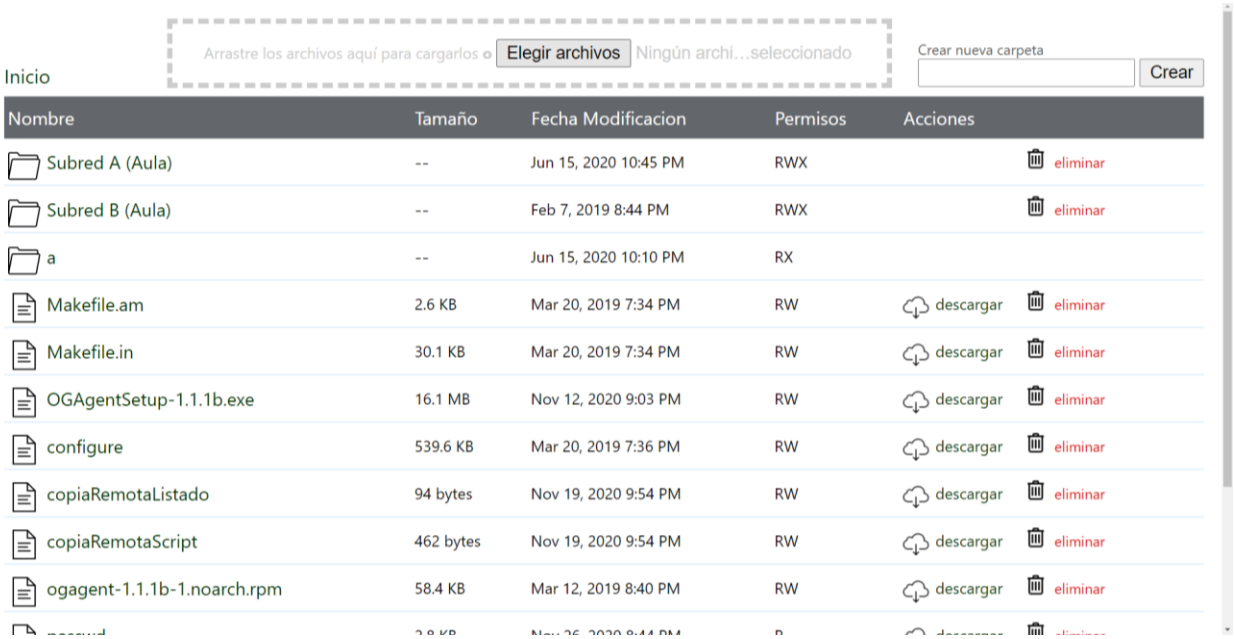

Ilustración 4-11 Página de gestor de ficheros

- Se podrá acceder a la página de gestión de conmutador.

Ilustración 4-12 Página de conmutador

4.2.9 Modificación de la página aula de la web de gestión de Opengnsys

Mediante esta modificación se quiere disponer de una forma sencilla de lanzar diferentes operaciones sobre varios equipos de una vez. Las operaciones que se podrán realizar tras la modificación serán:

- Encendido remoto de equipos.
- Apagar remoto de equipos.
- Enviar archivos a cliente ogLive.
- Enviar comandos o script a los equipos tanto a cliente ogLive como cliente ogAgent en los sistemas operativos
- Enviar procedimiento para la ejecución en cliente ogLive.

Una vez enviada las diferentes operaciones esta será guarda en la cola de acciones y de esta manera poder saber cuándo se realizó la acción.

A continuación, se indicará los diferentes archivos que se han modificados o se han creado y el porqué de estas modificaciones, partiremos de que la ruta inicial es /opt/opengnsys/www si se realicen la modificación tras la instalación de opengnsys o opengnsys /admin/WebConsole/ en caso de que se realicen sobre el código fuente

de Opengnsys.

- El archivo aula.php se para añadir en las funciones definida en el archivo personalizado/modificador_aula.php y realizar las llamadas a las funciones añadirDependencias, añdirInicioModificacion, añdirOpcionesModificacion, añdirFinModificacion se modificará la función pintaordenadores que realiza la presentación de los equipos de un aula para poder realizar las operaciones que anteriormente mencionadas. En caso de realizar la instalación también habría que modificar principal/aula.device.php.
- Se modifican los ficheros de idiomas/javascripts/xxx/aulas_xxx.js e idiomas/php/xxx/aulas_xxx.php para añadir las traducciones de los nuevas operaciones y elementos nuevos usados en tras la modificación. El valor de xxx variará entre esp, eng y cat dependiendo del idioma.
- Se crea el archivo personalizado/seleccion_fichero.php que con el cual se presentará una página con la cual se permitirá al usuario seleccionar los ficheros a enviar al cliente ogLive, las particiones donde realizar las copias de estos ficheros y la ruta dentro de las particiones seleccionadas.

  Es esta página contendrá la lógica para generar el script que será usado en el cliente ogLive para realizar la copia de los ficheros.

Este será configurado mediante los ficheros de configuración personalizado/configuracion/configuracion_fichero.php y personalizado/configuracion/mapeo_particiones_opengnsys.php, estos seran detallados en el

- Se crean los ficheros de idiomas/php/xxx/seleccion_ficheros_xxx.php que establecerán las traducciones necesarias para el fichero personalizado/seleccion_fichero.php. El valor de xxx variará entre esp, eng y cat dependiendo del idioma.
- Se crea el archivo personalizado/modificador_aula.php donde que contendrá las funciones necesarias para modificar aula.php con el fin de añadir las acciones que se quieren poder realizar:
    - añadirDependencias: Se encargará de añadir los archivos modificacionAula.css y modificacionAula.js que será usado para añadir estilo y la lógica necesaria a los elementos que se añadirá a cada equipo. Tambien añadirá como variables globales para javascript las traducciones necesarias y los datos sobre particiones y sistemas operativos para uso en la pantalla de enviar comandos.
    - añadirInicioModificacion: Esta recibe como argumento el nombre del aula e introduce el comienzo de un formulario que será usado para posteriormente controlar las acciones seleccionada.
    - añadirOpcionesModificacion: Esta recibe como argumentos el nombre del equipo, dirección MAC del equipo, dirección IP del equipo y la variable con las traducciones. Esta función añadirá debajo de cada equipo una serie de interruptores mediante los cuales se permitirá seleccionar una de las diferentes operaciones.
    - añadirFinModificacion: Esta recibe como argumentos el nombre del aula y la variable con las traducciones. Esta función añadirá los siguientes elementos:
        - Un selector para permitir elegir una misma operación para todos los equipos.
        - Un selector para permitir elegir el procedimiento a enviar.
        - Un botón que permitirá abrir la página seleccion_fichero.php.
        - Un botón para enviar las operaciones seleccionadas.
        - Un botón para eliminar las operaciones seleccionadas.
        - Un botón para ocultar o mostrar las operaciones.

      Para finalizar añadir el final del formulario que se inició con la función añadirInicioModificacion.
- Se crea el archivo personalizado/modificacionAula.js que incluye la lógica necesaria para realizar las siguientes acciones:
    - Controlar las operaciones seleccionadas, lanzar una alerta para solicitar la confirmación y enviarlas a la página personalizado/capturador_opciones_modificacion_aula.php al pulsar el botón de envió de operaciones.
    - Mostrar la ventana para permitir introducir el script a enviar y sobre qué sistema operativo aplicarlo en caso de que unas de las operaciones seleccionadas sea enviar comando.
    - Recibir el resultado de la página personalizado/capturador_opciones_modificacion_aula.php y generar las peticiones necesarias a la página /comandos/gestores/gestor_Comandos.php para que se realicen las acciones seleccionadas.
    - Procesar el script que se quiere enviar para hacer uso de lo configurado en personalizado/configuracion/mapeo_particiones_opengnsys.php y de esta manera genera el script que realmente será ejecutado en cliente ogLive o ogAgent.
    - Limitar que solo se pueda seleccionar una acción por cada equipo.
- Se crea el archivo personalizado/modificacionAula.css que incluye los estilos usado en la modificación.
- Se crea el archivo personalizado/capturador_opciones_modificacion_aula.php que se encargará de recibir las operaciones y equipos seleccionados y buscar para estos equipos su identificador para poder

devolverlo y que se usado para generar la llamada que ejecute la acción que realmente se quiere realizar.

Las modificaciones realizadas sobre los distintos ficheros se detallarán en el

ANEXO I: Modificaciones y procedimientos de opengnsys. Y los ficheros nuevos será entregado junto con la memoria.

Tras realizar las modificaciones se puede observar los siguientes cambios el web de gestión:

- Se podrá seleccionar las diferentes operaciones sobre un aula o varias aulas.

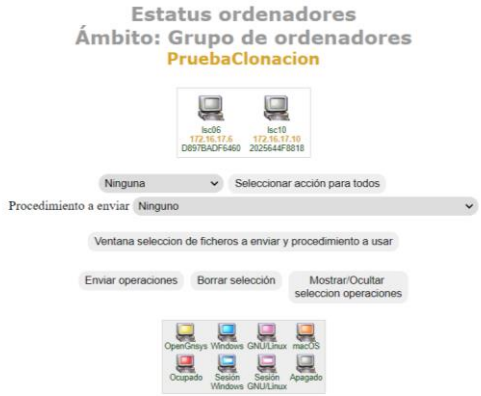

Ilustración 4-13 Aula con operaciones ocultas

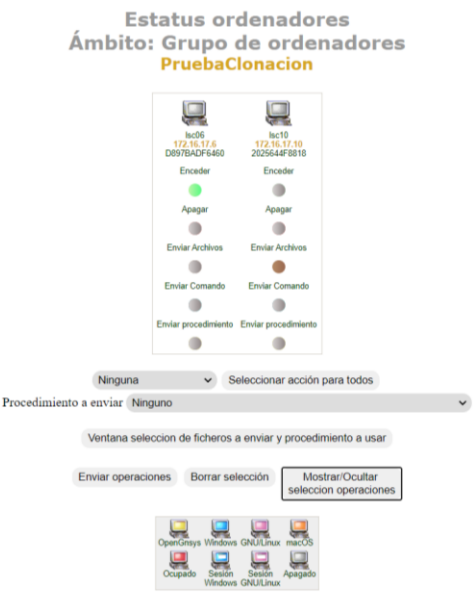

Ilustración 4-14 Aula con operaciones seleccionadas

- Se podrá seleccionar los diferentes archivos a enviar al cliente ogLive.

Ilustración 4-15 Pagina de selección de archivos a enviar

- Se podrá seleccionar introducir los comandos o script a enviar y sobre que sistemas operativos aplicar cuando se selecciona como operación a realizar enviar comando.

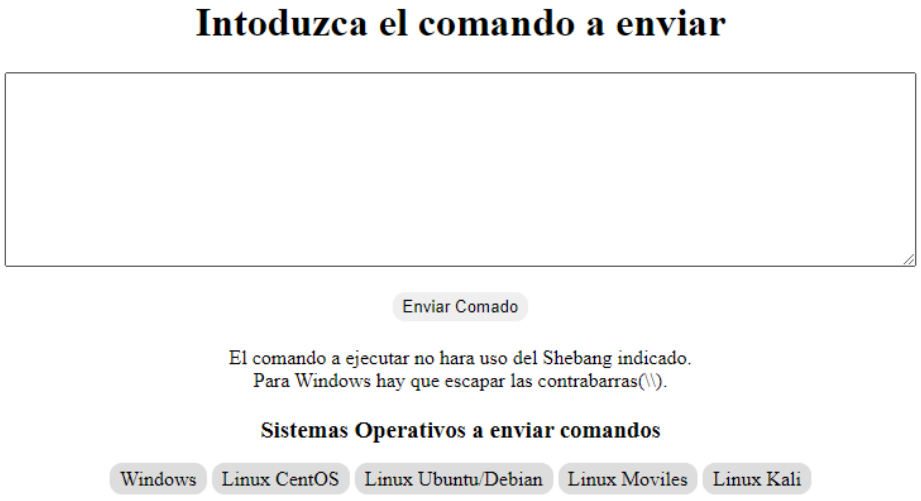

Ilustración 4-16 Página para introducir el comando a enviar

4.2.10 Líneas de continuación

Una de las posibles líneas de continuación podría ser la corrección de la limitación indicada en la corrección de para la restauración multicast.

Otra posible línea de continuación podría ser el cambio el funcionamiento de la página de sincronización entre Guacamole y Opengnsys para que solo se haga uso de sus respectivas API REST, esto permitirá que se mas mantenible y robusto a cambios en el modelo de datos de ambas aplicaciones.

Otra posible línea de continuación podría ser la adaptación de la web de gestión de Opengnsys para permitir el acceso a un Operador y que este solo pueda únicamente realizar algunas acciones.

Por último, otra posible línea de continuación se puede la integración del control remoto mediante Guacamole para que pueda ser llamada directamente desde la web de gestión de Opengnsys.

4.3 Control remoto mediante Guacamole

En esta sección explicará el proceso de instalación y configuración de la herramienta Apache Guacamole. Este se compone de una aplicación web que permitirá el acceso y control de los equipos desde cualquier navegador web sin la necesidad de instalar ningún cliente y una aplicación encargada de realizar la conexión a las clientes mediante VNC, RDP o SSH y proporcionar una interfaz usada por la aplicación web para el control de esta conexión.

4.3.1 Situación actual y problemática

En el laboratorio de Telemática se quiere poder permitir el control remotamente de los equipos.

Entre las acciones principales que se quieres poder realizar están:

- Permitir ceder el control de un equipo a un alumno desde del exterior.
- Tomar el control de equipo momentánea para realizar unos cambios o resolver alguna duda.

Actualmente en el laboratorio se están haciendo uso de las herramientas Veyon[104] y Anydesk[105].

Veyon permite realizar un control de aula de forma sencilla, permitiendo el encendido, apagado o control remoto entre otras funcionalidades. Entre las desventajas encontradas para esta herramienta son que carece de un método para poder ceder el control de un equipo a un usuario, no puede realizarse el control a través de Internet y requiere de la instalación de una herramienta especifica tanto en el equipo a controlar como en el cliente.

Por otro lado, Anydesk permite el control remoto a través de internet incluso si su dirección IP cambia. Entre las desventajas encontradas para esta herramienta son que carece un método que permita cambiar el acceso entre diferentes usuarios y requiere la instalación de una herramienta especifica tanto en el equipo a controlar como en el cliente.

Se aptado por el uso de Guacamole para abordar algunas de las carencias de las herramientas anteriormente menciana debido a que Guacamole permite la creación de usuarios y la asignación de diferentes conexiones a estos usuarios, dispone de una aplicación web que permite el control de los equipos desde un navegador y hace uso de los protocolos VNC, RDP y SSH sin necesidad de una herramienta especifica en el equipo a controlar.

Además, se ha creado una página que permite la sincronización de los equipos de Opengnsys sobre Guacamole y de esta manera crear la configuración de las conexiones a los equipos del laboratorio de forma automatizada. Esto es detallado en la sección Modificación de la barra superior de la web de gestión de Opengnsys.

4.3.2 Solución elegida

El escenario propuesto para detallará la solución optada tendrá las siguientes características:

- El sistema operativo usado será Debian 10.
- Se hará uso de la una base de datos basado en mysql.
- Se hará la redirección del tráfico para conectar con Apache Tomcat[106] a través de Apache HTTP Server[107].
- Partiremos de que no se encuentra instalado el servidor Apache Tomcat y que Apache HTTP Server ya se encuentra instalado y en uso.

Para finalizar el proceso se realizará sobre la última vesion estable existe v1.2.0, esto provoca que los links de descarga del código fuente, archivo war y librerías sean específico para esta versión y han sido obtenido de la directamente de la página de descarga el proyecto[108].

Para realizar la instalación se ha de seguir lo siguientes pasos como superusuario:

1. Instalar de las dependencias necesarias y además Apache Tomcat. Estas dependencias son detalladas en el manual de Guacamole, serán instalada las dependencias tantos la obligatorias como la opcionales. Para ello haremos uso de la herramienta apt.

```
 apt -y install libcairo2-dev libjpeg62-turbo-dev libpng12-dev
libtool-bin libossp-uuid-dev libavcodec-dev libavformat-dev libavutil-
dev libswscale-dev freerdp2-dev libpango1.0-dev libssh2-1-dev
libtelnet-dev libvncserver-dev libwebsockets-dev libpulse-dev libssl-
dev libvorbis-dev libwebp-dev default-jdk tomcat8 tomcat8-admin
tomcat8-common tomcat8-docs tomcat8-user
```

Los argumentos usados son:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
…	Indica todos los paquetes a instalar.

2. Descargar el código fuente de servidor guacd para ello haremos uso de la herramienta wget.

```
 wget -O guacamole-server-1.2.0.tar.gz
"http://apache.org/dyn/closer.cgi?action=download&filename=guacamole/
1.2.0/source/guacamole-server-1.2.0.tar.gz"
```

El argumento usado es:

-O guacamole-server-1.2.0.tar.gz	Estable el nombre del fichero tras realizar la descarga.
http://apache.org/dyn/closer.cgi?action=download&filename=guacamole/1.2.0/source/guacamole-server-1.2.0.tar.gz	Indica la URL donde se ubica el fichero a descargar.

3. Descomprimir el archivo descargado para ello haremos uso de la herramienta tar.

```
 tar -xzvf guacamole-server-1.2.0.tar.gz
```

Los argumentos usados son:

-x	Indica que se va a realizar la operación de descompresión.
-v	Indica que se realice en modo verboso y de esta manera mostrar lo archivos que han sido descomprimido.
-z	Establece que el formato del fichero a descomprimir es gzip.
-f	Indica que el origen de los datos a descomprimir es un fichero.
guacamole-server-1.2.0.tar.gz	Indica el fichero a descomprimir.

4. Acceder al directorio descomprimido para ello haremos uso de la herramienta cd.

```
cd guacamole-server-1.2.0
```

El argumento usado es:

guacamole-server-1.2.0	Indica el directorio al que se accederá.

5. Lanzar la configuración de las dependencias para ello lanzaremos el script configure.

```
./configure --with-init-dir=/etc/init.d
```

El argumento usado es:

--with-init-dir=/etc/init.d	Indica el directorio donde colocar el script de servicio con el fin de que el demonio guacd se inicie automáticamente.

6. Lanzar la compilación e instalación de guacd para ello se hará uso de herramienta make[109].

```
make
make install
```

El argumento usado es:

install	Indica que se realice la instalación de los elementos compilado anterior mente.

7. Actualizar la cache del sistema con las nuevas librerías instaladas para ello se hará uso de la herramienta ldconfig[110].

```
ldconfig
```

8. Volver al directorio superior para ello se hará uso de la herramienta cd.

```
cd -
```

El argumento usado es:

-	Indica que se vuelva al directorio desde el cual se accedió al directorio actual.

9. Descargar el fichero war de la aplicación web cliente de Guacamole para ello haremos uso de la herramienta wget.

```
wget -O guacamole-1.2.0.war
"http://apache.org/dyn/closer.cgi?action=download&filename=guacamole/
1.2.0/binary/guacamole-1.2.0.war"
```

Los argumentos usados son:

-O guacamole-1.2.0.war	Estable el nombre del fichero tras realizar la descarga.
http://apache.org/dyn/closer.cgi?action=download&filename=guacamole/1.2.0/binary/guacamole-1.2.0.war	Indica la URL donde se ubica el fichero a descargar.

10. Copiar el fichero war en la carpeta de Tomcat para que sea desplegado automáticamente para ello haremos uso del comando cp.

```
cp guacamole-1.2.0.war /var/lib/tomcat8/webapps/guacamole.war
```

Los argumentos usados son:

guacamole-1.2.0.war	Indica el fichero a copiar.
/var/lib/tomcat8/webapps/guacamole.war	Indica el fichero destino mediante su ruta absoluta. Es importante el cambio de nombre ya que será usado por Tomcat para indicar la ruta de acceso a la aplicación.

11. Crear los directorios necesarios para configurar la librería para la autentificación mediante la base de datos. Para ello se hará uso de la herramienta mkdir.

```
mkdir /etc/guacamole/lib
mkdir /etc/guacamole/extensions
```

El argumento usado es:

/etc/guacamole/lib /etc/guacamole/extensions	Indica el directorio a crear mediante su ruta absoluta.

12. Descargar el fichero con la librería para la autentificación mediante la base de datos para ello haremos uso de la herramienta wget.

```
wget -O guacamole-auth-jdbc-1.2.0.tar.gz
"http://apache.org/dyn/closer.cgi?action=download&filename=guacamole/
1.2.0/binary/guacamole-auth-jdbc-1.2.0.tar.gz"
```

Los argumentos usados son:

-O guacamole-auth-jdbc-1.2.0.tar.gz	Estable el nombre del fichero tras realizar la descarga.
http://apache.org/dyn/closer.cgi?action=download&filename=guacamole/1.2.0/binary/guacamole-auth-jdbc-1.2.0.tar.gz	Indica la URL donde se ubica el fichero a descargar

13. Descomprimir el archivo descargado para ello haremos uso de la herramienta tar.

```
tar -xzvf guacamole-auth-jdbc-1.2.0.tar.gz
```

Los argumentos usados son:

-x	Indica que se va a realizar la operación de descompresión.
-v	Indica que se realice en modo verboso y de esta manera mostrar lo archivos que han sido descomprimido.
-z	Establece que el formato del fichero a descomprimir es gzip.
-f	Indica que el origen de los datos a descomprimir es un fichero.
guacamole-auth-jdbc-1.2.0.tar.gz	Indica el fichero a descomprimir.

14. Copiar la librería para mysql para que pueda ser usada por Guacamole para ello haremos uso del comando cp.

```
cp guacamole-auth-jdbc-1.2.0/mysql/guacamole-auth-jdbc-mysql-1.2.0.jar /etc/guacamole/extensions/guacamole-auth-jdbc-mysql.jar
```

Los argumentos usados son:

guacamole-auth-jdbc-1.2.0/mysql/guacamole-auth-jdbc-mysql-1.2.0.jar	Indica el fichero a copiar.
/etc/guacamole/extensions/guacamole-auth-jdbc-mysql.jar	Indica el fichero destina mediante su ruta absoluta.

15. Descargar el conector JBDC para la base de datos necesario para ello haremos uso de la herramienta wget.

```
wget -O mysql-connector-java-5.1.40.tar.gz "http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.40.tar.gz"
```

Los argumentos usados son:

-O mysql-connector-java-5.1.40.tar.gz	Estable el nombre del fichero tras realizar la descarga.
http://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.40.tar.gz	Indica la URL donde se ubica el fichero a descargar

16. Descomprimir el archivo descargado para ello haremos uso de la herramienta tar.

```
tar -xzvf mysql-connector-java-5.1.40.tar.gz
```

Los argumentos usados son:

-x	Indica que se va a realizar la operación de descompresión.
-v	Indica que se realice en modo verboso y de esta manera mostrar lo archivos que han sido descomprimido.
-z	Establece que el formato del fichero a descomprimir es gzip.
-f	Indica que el origen de los datos a descomprimir es un fichero.
mysql-connector-java-5.1.40.tar.gz	Indica el fichero a descomprimir.

17. Copiar la librería para mysql para que pueda ser usada por Guacamole para ello haremos uso del comando cp.

```
cp mysql-connector-java-5.1.40/mysql-connector-java-5.1.40-bin.jar
/etc/guacamole/lib/mysql-connector-java-5.1.40-bin.jar
```

Los argumentos usados son:

mysql-connector-java-5.1.40/mysql-connector-java-5.1.40-bin.jar	Indica el fichero a copiar.
/etc/guacamole/lib/mysql-connector-java-5.1.40-bin.jar	Indica el fichero destina mediante su ruta absoluta.

18. Modificar el fichero de configuración de guacd para incluir los parámetros de configuración conexión con la base de datos. El contenido del fichero /etc/guacamole/guacamole.properties deberá ser el siguiente:

```
guacd-hostname: localhost
guacd-port:     4822
mysql-hostname: localhost
mysql-port: 3306
mysql-database: guacamole_db
mysql-username: guacamole_userdb
mysql-password: guacamole_passwordb
```

Los parámetros de configuración son:

guacd-hostname	Indica la dirección donde se ubica el demonio guacd.
guacd-port	Indica el puerto donde se encuentra el demonio guacd.
mysql-hostname	Indica la dirección donde se ubica la base datos.
mysql-port	Indica el puerto donde se encuentra la base de datos.
mysql-database	Indica el nombre de la base de datos destinada a Guacamole.
mysql-username	Indica el nombre de usuario de acceso a la base de datos de Guacamole.
mysql-password	Indica la contraseña del usuario de acceso a la base de datos de Guacamole.

19. Crear la base de datos y el usuario de acceso destinado para Guacamole. Para realizar esto será necesario ejecutar la siguientes secuencia SQL

```
 CREATE DATABASE guacamole_db;
 CREATE    USER    'guacamole_userdb'@'localhost'    IDENTIFIED    BY
'guacamole_passwordb';
 GRANT    SELECT,INSERT,UPDATE,DELETE    ON    guacamole_db.*    TO
'guacamole_userdb'@'localhost';
 FLUSH PRIVILEGES;
```

Estas sentencias crean la base datos con nombre guacamole_db, crea el usuario guacamole_userdb con la contraseña guacamole_passwordb, le asigna los permisos necesarios y limita su acceso únicamente desde la propia máquina.

20. Crear las tablas y datos necesario en la base de datos de Guacamole mediante el script descargado con la librería de autentificación mediante base de datos. Para ello haremos uso las herramientas msyql.

```
 mysql -uroot -prootpassword guacamole_db < guacamole-auth-jdbc-
1.2.0/mysql/schema/001-create-schema.sql
 mysql -uroot -prootpassword guacamole_db < guacamole-auth-jdbc-
1.2.0/mysql/schema/002-create-admin-user.sql
```

Los argumentos usados son:

-uroot	Indica que se hará uso del usuario root.
rootpassword	Indica la contraseña del usuario anteriormente indicado.
ogAdmBD	Indica el nombre de la base datos de Opengnsys y sobre la que se aplicará el script.
guacamole-auth-jdbc-1.2.0/mysql/schema/001-create-schema.sql guacamole-auth-jdbc-1.2.0/mysql/schema/002-create-admin-user.sql	Indica el nombre del script de sql que se va a aplicar.

El script guacamole-auth-jdbc-1.2.0/mysql/schema/002-create-admin-user.sql creará un primer usuario llamada guacadmin y contraseña guacadmin con todos los permisos disponibles y con el cual se podrá iniciar y empezar

a configurar el cliente web de Guacamole.

21. Configurar el servidor Apache Tomcat para recibir la redireccion de Apache HTTP Server. Para ello se añadirá el siguiente fragmento en el fichero de configuración /var/lib/tomcat8/conf/server.xml

```
  <Connector port="8081" protocol="HTTP/1.1"
         connectionTimeout="20000"
         proxyPort="80" />
    <Connector port="8082"
          protocol="org.apache.coyote.http11.Http11AprProtocol"
          secure="true" scheme="https"
          SSLEnabled="true"
          SSLCertificateFile="/etc/ssl/apache/apache.crt"
          SSLCertificateKeyFile="/etc/ssl/private/ssl-cert-
snakeoil.key"
          SSLCACertificateFile="/etc/ssl/certs/ssl-cert-snakeoil.pem"
          SSLVerifyClient="require"
          SSLVerifyDepth="10"
          clientAuth="true"
          proxyPort="443" />
```

Este fragmento permite recibir tráfico desde el puerto 80 y 443 y se hará uso de los mismos certificados configurado por defecto en apache, si se hubiera cambiado será necesario cambiar los parámetros SSLCertificateFile, SSLCertificateKeyFile y SSLCACertificateFile.

22. Crear la configuración para que realizar la redireccion de Guacamole desde Apache HTTP Server a Apache Tomcat, se redireccionará tanto el acceso web como el acceso al websocket que es vital para el óptimo funcionamiento del cliente. Para ello se debe crear el fichero /etc/apache2/sites-available/guacamole.conf con el siguiente contenido:

```
  <Location /guacamole/>
      Order allow,deny
      Allow from all
      ProxyPass http://localhost:8081/guacamole/ flushpackets=on
      ProxyPassReverse http://localhost:8081/guacamole/
  </Location>

 <Location /guacamole/websocket-tunnel>
       Order allow,deny
       Allow from all
       ProxyPass ws://localhost:8081/guacamole/websocket-tunnel
       ProxyPassReverse     ws://localhost:8081/guacamole/websocket-
tunnel
  </Location>
```

23. Activar los módulos de Apache HTTP Server necesarios para realizar la redirección. Para ello se hará uso de la herramienta a2enmod[111].

```
a2enmod proxy
a2enmod proxy_httpt
a2enmod proxy_connect
a2enmod proxy_wstunnel
```

El argumento usado es:

proxy proxy_httpt proxy_connect proxy_wstunnel	Indica el módulo que se quiere habilitar.

24. Activar la configuración que realizará la redirección. Para ello se hará uso de la herramienta a2ensite[112].

```
a2ensite guacamole.conf
```

El argumento usado es:

guacamole.conf	Indica el nombre de la configuración a activar.

25. Reiniciar el servidor Apache HTTP Server para que tenga efecto los cambios realizado. Para ello se usará de la herramienta systemctl.

```
systemctl restart apache2.service
systemctl enable tomcat8.service
```

Los argumentos usados serán:

enable	Establece que el servicio se reiniciará.
apache2.service	Indica el nombre del servicio que se reiniciar.

26. Habilitar para que se autoarranque los servicios de guacd y tomcat al iniciar. Para ello se usará de la herramienta systemctl.

```
systemctl enable guacd.service
systemctl enable tomcat8.service
```

Los argumentos usados serán:

enable	Establece que el servicio se autoarranque al iniciar el sistema.
guacd.service tomcat8.service	Indica el nombre del servicio que se desea configurar.

27. Iniciar los servicios de guacd y tomcat. Para ello se usará de la herramienta systemctl.

```
systemctl start guacd.service
systemctl start tomcat8.service
```

Los argumentos usados serán:

start	Indica que se inicie un servicio.
guacd.service tomcat8.service	Indica el nombre del servicio que se desea iniciar.

En el

ANEXO J: Control remoto mediante guacamole se detallará como realizar la configuración de los servidores de VNC, SSH o RDP y en los sistemas operativos.

4.3.3 Líneas de continuación

Una posible línea de continuación es el estudio de cómo realizar la integración con la autentificación usada en la universidad y de esta manera poder acceder al sistema mediante el uso del uvus.

Otra posible línea de continuación es el estudio completo de todas las características ofrecida por Guacamole en cuanto a parámetro para establecer la conexión ya que estos van desde compartir el portapapeles hasta la transferencia de archivo.

Otra posible línea de continuación es el estudio de la herramienta que está en desarrollo OpenRLabs cuyo objetivo es el uso de los equipos administrador por Opengnsys a través de web y cuyo backend está formado por el demonio de Guacamole guacd.

5 GESTIÓN DE LA RED INTERNA

En este capítulo se detallará soluciones que permitirá el acceso remoto red interna del laboratorio de Telemática, automatizar tareas de gestión de los conmutadores y un plan de contingencia antes fallo en la red interna del laboratorio de Telemática.

5.1 Acceso mediante VPN

En esta sección detallará el proceso de instalación y configuración de un servidor VPN basado en OpenVPN[113]. Este permitirá el acceso a las subredes del laboratorio y poder trabajar como un equipo más de una de estas subredes.

Cuando se hace referencia a una conexión VPN (sigla de Virtual Private Network) se trata de una conexión que permite el acceso a una red privada desde el exterior de esta y navegar y acceder a los recursos de esta como si se estuviera conectado a esta red privada.

5.1.1 Situación actual y problemática

En el laboratorio de Telemática se dispone de algunos servicios ofrecidos por los servidores que no son accesible desde el exterior de la red interna debido a los puertos los cuales pueden ser expuesto al exterior, estos son administrado por la red interna de la universidad de Sevilla.

Además, se quiere poder permitir el acceso a alumnado a la red interna con el fin de que puedan hacer uso de algunos elementos de la red interna del laboratorio.

Para permitir el acceso remoto a la red del laboratorio se hará uso de red VPN que permita a los usuarios de los mismo conectarse como un equipo más dentro de una de las subredes de este, para ello se hará que el cliente al conectarse solicite la configuración al servidor DHCP.

5.1.2 Solución elegida

El escenario propuesto para detallará la solución optada tendrá las siguientes características:

- El sistema operativo en el servidor usado será Debian 10.
- Se dispondrá de dos subredes distintas que compondrá la red del laboratorio. La subred A será accesible mediante la interfaz eth1 y la subred B será accesible mediante la interfaz eth2.

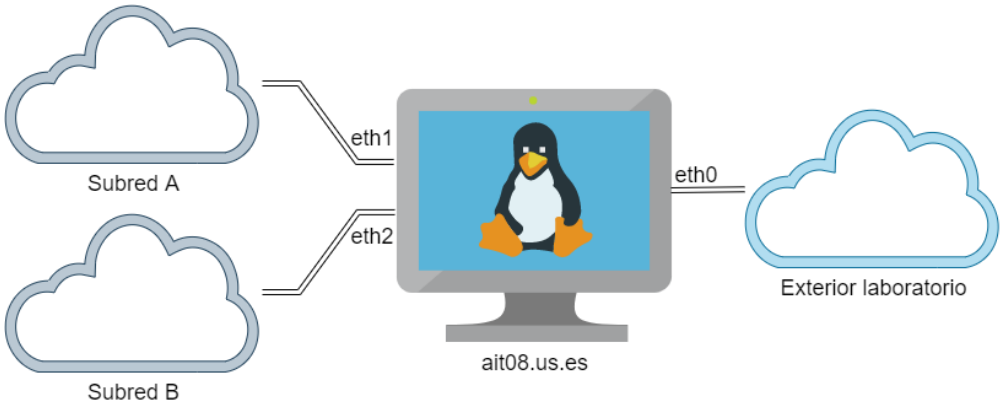

Ilustración 5-1 Esquema de enrutamiento simplificado del laboratorio de Telemática

En la solución propuesta se creará dos servidores VPN, uno para cada subred, y los clientes que accedan a estos servidores solicitará su dirección IP al servidor DHCP que se encuentre activo en la red interna del laboratorio.

5.1.2.1 Instalación y configuración de los servidores VPN

Se detallará el proceso para realizar la configuración inicial del servidor VPN. Para ellos se debe seguir los siguientes pasos como superusuario:

1. Instalar el servidor VPN OpenVPN y las dependencias necesarias para la creación de los certificados y claves necesarios para establecer la conexión. Para ello haremos uso de la herramienta apt.

```
apt -y install openvpn easy-rsa
```

Los argumentos usados son:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
openvpn easy-rsa	Indica todos los paquetes a instalar.

2. Copiar el directorio de easy-rsa a la carpeta de configuración de OpenVPN para ser usado posteriormente para la creación de los certificados y claves. Para ello haremos uso de la herramienta cp.

```
cp -r /usr/share/easy-rsa/ /etc/openvpn
```

Los argumentos usados son:

-r	Indica que se haga la copia del contenido del origen de manera recursiva.
/usr/share/easy-rsa/	Indica del directorio a copiar.
/etc/openvpn	Indica el fichero destino.

3. Crear el directorio que contendrá la claves. Para ello se hará uso de la herramienta mkidr.

```
mkdir /etc/openvpn/easy-rsa/keys
```

El argumento usado es:

/etc/openvpn/easy-rsa/keys	Indica el directorio a crear mediante su ruta absoluta.

4. Editar el fichero de las variables necesaria para la creación de los certificados de los servidores VPN. El contenido del fichero /etc/openvpn/easy-rsa/vars a editar serán las siguientes variables:

```
export KEY_COUNTRY="ES"
export KEY_PROVINCE ="SE"
export KEY_CITY="Sevilla"
export KEY_ORG="LabTelematica"
export KEY_EMAIL="x@x.us.es"
export KEY_OU="ETSI"
export KEY_NAME="ait08"
```

Las variables que son modificadas son:

KEY_COUNTRY	Indica el país.
KEY_PROVINCE	Indica la provincia.
KEY_CITY	Indica la ciudad.
KEY_ORG	Indica el nombre de la organización.
KEY_EMAIL	Indica un emai. Este puede ser el de administrador u otro email representativo.
KEY_OU	Indica la unidad organizacional.
KEY_NAME	Indica el nombre del servidor.

5. Crear el enlace simbólico a la versión de openssl que se quiera. Para ello se hará uso de la herramienta ln.

```
ln -fs /etc/openvpn/easy-rsa/openssl-1.0.0.cnf /etc/openvpn/easy-rsa/openssl.cnf
```

Los argumentos usados son:

-f	Indica que si existiera ya el enlace en el destino se borre y se cree de nuevo.
-s	Indica que se cree un enlace simbólico.
/etc/openvpn/easy-rsa/openssl-1.0.0.cnf	Indica la ruta de origen para la cual se creará el enlace.
/etc/openvpn/easy-rsa/openssl.cnf	Indica destino donde crear el enlace.

6. Acceder al directorio /etc/openvpn/easy-rsa. Para ello se hará uso de la herramienta cd.

```
cd /etc/openvpn/easy-rsa
```

El argumento usado es:

etc/openvpn/easy-rsa	Indica el directorio al que se accederá.

7. Cargar las variables de entorno necesaria para crear los certificados. Para ello se hará uso de la herramienta source.

```
source ./vars
```

El argumento usado es:

./vars	Indica el fichero que será cargado.

8. Limpiar todos los certificados existentes por si los hubiese. Para ello se debe ejecutar el siguiente script:

```
./clean-all
```

9. Crear el certificado de autentificación. Para ello se debe ejecutar el siguiente script:

```
./build-ca
```

Al ejecutar el script se solicitará introducir una serie de datos que al haberse modificado y exportado vars ya se encontrará configurada con los valores adecuado y solo habrá que pulsar Intro.

10. Renombramos el certificado de autentificación creado para poder crear luego para el otro servidor OpenVPN. Para ello se hará uso de la herramienta mv.

```
mv keys/ca.crt keys/ca_ltA.crt
mv keys/ca.key keys/ca_ltA.key
```

Los argumentos usados son:

keys/ca.crt keys/ca.key	Indica el fichero origen.
keys/ca_ltA.crt keys/ca_ltA.key	Indica el fichero destino.

11. Crear los certificados para el servidor OpenVPN. Para ello se debe ejecutar el siguiente script:

```
./build-key-server server_ltA
```

El argumento usado es:

server_ltA	Indica el nombre se le asignará los certificados.

Los pasos del 9 al 11 es necesario realizarlo de nuevo cambiando ltA por ltB.Es necesario dos conjuntos de certificados para que poder diferenciar a que servidor pertenecen los clientes cuando se reciban las peticiones de conexiones.

12. Crear el archivo de establecimiento de claves Diffie-Helman. Para ello se debe de la herramienta openssl[114].

```
openssl dhparam -out /etc/openvpn/dh4096.pem 4096
```

Los argumentos usados son:

dhparam	Indica que se genere las claves Diffie-Helman.
-out /etc/openvpn/dh4096.pem	Indica el fichero donde se almacenará las claves generada.
4096	Indica el número de bits usado a la hora de generar las claves.

13. Crear la clave HMAC. Para ello se debe de la herramienta openvpn[115].

```
openvpn --genkey --secret /etc/openvpn/easy-rsa/keys/hmac.key
```

Los argumentos usados son:

--genkey	Indica que Genere una clave aleatoria para usar como secreto compartido.
--secret /etc/openvpn/easy-rsa/keys/hmac.key	Indica el fichero donde se almacenará la clave generada.

14. Crear las claves para un usuario que será usado únicamente para la configuración la verificación que permitirá conocer a que usuario fue revocado su acceso. Para ello se debe ejecutar el siguiente script:

```
./build-key clientRevocado
```

El argumento usado es:

clientRevocado	Indica el nombre que se le asignará a los certificados generados.

15. Revocar el acceso al usuario creado anteriormente. Para ello se debe ejecutar el siguiente script:

```
./revoke-full clientRevocado
```

El argumento usado es:

clientRevocado	Indica el nombre de los certificados a los cueles se le revocará el acceso.

16. Crear los ficheros de configuración para los servidores OpenVPN. Se crearán los siguientes ficheros:

```
/etc/openvpn/server_ltA.conf
/etc/openvpn/server_ltB.conf
```

El contenido de estos ficheros se detallará en el

ANEXO K: Acceso mediante VPN.

17. Recargar la configuración del administrador de systemd para que sea detectado los servidores OpenVPN. Para ello se hará uso de la herramienta systemctl.

```
systemctl daemon-reload
```

El argumento usado es:

daemon-reload	Indica que se inicie la recarga de.

18. Habilitar para que se autoarranque los servicios de cada servidor OpenVPN al iniciar. Para ello se usará de la herramienta systemctl.

```
systemctl enable openvpn@server_ltA.service
systemctl enable openvpn@server_ltB.service
```

Los argumentos usados serán:

enable	Establece que el servicio se autoarranque al iniciar el sistema.
openvpn@server_ltA.service openvpn@server_ltB.service	Indica el nombre del servicio que se desea configurar.

19. Habilitar para que se autoarranque el servicio configuración de los bridges al iniciar. Para ello se usará de la herramienta systemctl.

```
systemctl enable bridgeOpenvpn.service
```

Los argumentos usados serán:

enable	Establece que el servicio se autoarranque al iniciar el sistema.
bridgeOpenvpn.service	Indica el nombre del servicio que se desea configurar.

Este servicio será detallado completamente en el

ANEXO K: Acceso mediante VPN. Este servicio será el encargado de generar un bridge que cada para la interfaz de cada subred (eth1 y eth2), donde se también se ubicaran los dispositivos TAP que será usado por los servidores.

Los dispositivos TAP será donde se conectará los clientes y que permitirá acceder a la subred donde se encuentre este dispositivo.

20. Iniciar los servicios de los sevidores OpenVPN y de configuración de los bridges. Para ello se usará de la herramienta systemctl.

```
systemctl start openvpn@server_ltA.service
systemctl start openvpn@server_ltB.service
systemctl start bridgeOpenvpn.service
```

Los argumentos usados serán:

start	Indica que se inicie un servicio.
openvpn@server_ltA.service openvpn@server_ltB.service bridgeOpenvpn.service	Indica el nombre del servicio que se desea iniciar.

5.1.2.2 Creación de cliente VPN

Se detallará el proceso para realizar la configuración un cliente para la realizar la conexión a la subred A. Para ellos se debe seguir los siguientes pasos como superusuario:

1. Copiar la configuración base para la subred A. Para ello se hará uso de la herramienta cp.

```
cp /etc/openvpn/client/cliente_ltA.conf
/etc/openvpn/client/cliente_nuevo.conf
```

Los argumentos usados son:

/etc/openvpn/client/cliente_ltA.conf	Indica el fichero origen de la copia.
/etc/openvpn/client/cliente_nuevo.conf	Indica el fichero destino de la copia

El contenido del fichero de configuración base /etc/openvpn/client/cliente_ltA.conf se detallará en el anexo

ANEXO K: Acceso mediante VPN

2. Acceder al directorio /etc/openvpn/easy-rsa. Para ello se hará uso de la herramienta cd.

```
cd /etc/openvpn/easy-rsa
```

El argumento usado es:

etc/openvpn/easy-rsa	Indica el directorio al que se accederá.

3. Cargar las variables de entorno necesaria para crear los certificados. Para ello se hará uso de la herramienta source.

```
source ./vars
```

El argumento usado es:

./vars	Indica el fichero que será cargado.

4. Crear los certificados para el cliente OpenVPN. Para ello se debe ejecutar el siguiente script:

```
./build-key cliente
```

El argumento usado es:

cliente_nuevo	Indica el nombre se le asignará los certificados.

5. Copiar los certificados al final de la configuración del cliente. Para ello se debe ejecutar el siguiente conjunto de acciones que cuya finalidad es añadir la etiqueta ca, cert y key en el fichero de configuración.

```
 echo '<ca>' >> /etc/openvpn/client/cliente_nuevo.conf
 cat             /etc/openvpn/easy-rsa/keys/ca_ltA.crt             >>
/etc/openvpn/client/cliente_nuevo.conf
 echo '</ca>' >> /etc/openvpn/client/cliente_nuevo.conf
 echo '<cert>' >> /etc/openvpn/client/cliente_nuevo.conf
 cat         /etc/openvpn/easy-rsa/keys/cliente_nuevo.crt         >>
/etc/openvpn/client/cliente_nuevo.conf
 echo '</cert>' >> /etc/openvpn/client/cliente_nuevo.conf
 echo '<key>' >> /etc/openvpn/client/cliente_nuevo.conf
 cat         /etc/openvpn/easy-rsa/keys/cliente_nuevo.key         >>
/etc/openvpn/client/cliente_nuevo.conf
 echo '</key>' >> /etc/openvpn/client/cliente_nuevo.conf
```

6. Crear el fichero de configuración específica para el cliente. Para la solución este fichero estará vacío. Para ello se creará el siguiente fichero:

```
/etc/openvpn/client/cliente_nuevo
```

De no existir este fichero el cliente no tendrá acceso. Esto puede ser de utilidad para limitar el acceso momentáneamente.

Para crear un cliente para la subred B únicamente habría que sustituir en el proceso ltA por ltB.

Si se quiere hacer uso de la configuración con un cliente grafico de OpenVPN solo será necesario remombrar el fichero cliente_nuevo.conf como cliente_nuevo.ovpn.

Para finalizar se debe tener en cuenta que debida al uso de la interfaz TAP para realizar la conexión no será posible conectarse desde dispositivos Android o IOS, esto se debe a que actualmente no es soportado de forma nativa por los clientes de OpenVPN este tipo de interfaz.

5.1.3 Líneas de continuación

Una posible línea de continuación es el estudio de un error detectado en las pruebas y es que dos clientes de diferentes servidores OpenVPN no puede comunicarse entre ellos.

Otra posible línea de continuación es la creación de un mecanismo para la creación nuevos clientes y revocación de clientes automatizado y de uso mediante una interfaz gráfica.

Otra posible línea de continuación puede ser la mejora la manera de configurar los bridges y simplificar los scripts necesarios o realizar la configuración de manera estática.

5.2 Scripts de gestión de conmutadores

En esta sección se detallarán como hacer uso de una serie de script que permitirán que permitirán realizar algunas funciones básicas para la gestión de los conmutadores que actualmente se usan en el laboratorio como elementos de la red interna.

Estos scripts permitirán la gestión mediante puerto serie o mediante ssh.

5.2.1 Situación actual y problemática

En el laboratorio de Telemática se utilizan varios modelos diferentes de conmutadores para formar la red interna del laboratorio, de ellos se quieren facilitar la gestión de los siguientes modelos:

- HP Procurve 2620-25
- HP Procurve 2650-25
- Cisco 3750G-24TS

Actualmente para realizar las tareas todas las tareas de gestión con estos se realizan introduciendo los comandos directamente en la terminal del puerto serie.

Esto implica que las personas que realizan estas tareas deben conocer los comando y sentencias de concreta para cada modelo de conmutador y se requiere una conexión directa entre el equipo desde el que se realiza las tareas y el conmutador afectado mediante un cable serie, estos pueden ser complicado por la distancia entre los equipos e incluso por la inexistencia del puerto serie.

Para ello se han ideado una serie de script cuya función es automatizar la realización de las siguientes tareas:

- Realizar una copia de la configuración actual del conmutador.
- Restaurar una copia de la configuración del conmutador.
- Realizar la carga de un firmware en el conmutador.
- Deshabilitar el servicio de DHCP del conmutador.
- Habilitar el servicio de DHCP del conmutador.
- Habilitar la web de gestión del conmutador.

El script ideado de dividen entre los que hacen uso del puerto serie y aquellos que hacen uso de SSH.

5.2.2 Script para la gestión de conmutadores basado en el uso de puerto serie

A continuación, se detallará el uso y funcionamiento de una serie de script basado en el uso de la comunicación serie con el conmutador mediante la herramienta minicom[116] en sistema operativos basado en Linux. Estos deberán ser ejecutado superusuario.

Para el uso de los scripts tiene las siguientes dependencias:

- Minicom, herramienta con la que se realizará la conexión serie y que ejecutará los scripts con los comandos especifico de cada conmutador.
- Sz[117], herramienta con la que se realizará él envió de fichero mediante el protocolo XMODEM.
- Rx[118], herramienta con la que se recibirán fichero mediante el protocolo XMODEM.
- Sed, herramienta que será usada para adaptar los scripts en función a los parámetros introducido.

Además, para los scripts para el conmutador Cisco 3750G-24TS será necesario las siguientes dependencias adicionales:

- Ip, herramienta que será usada para crear una nueva dirección IP local para enviar y recibir ficheros mediante el protocolo TFTP.
- In.tftpd[119], demonio usado para enviar y recibir ficheros mediante el protocolo TFTP, este debe encontrarse activo.

Tambien será necesario conocer el dispositivo asociado a la conexión serie, este deberá ser algo como /dev/ttySX, y es recomendable conocer la velocidad en baud a la que se encuentra configurado el conmutador, de no conocerlo es posible probar varios en ejecuciones consecutiva.

Para estos scripts el funcionamiento se diferencia entre los modelos de HP y de CISCO para todos los modelos de los conmutadores.

Para la ejecución de los scripts se hace uso de la contraseña que habilita privilegios para modificar la configuración del conmutador. Esta contraseña es aquella que se solicita al usar el siguiente comando:

```
enable
```

El procedimiento para realizar configuración de esta contraseña se detallará en el ANEXO L: Scripts de gestión de conmutadores.

5.2.2.1 Conmutador Cisco 3750G-24TS

Se dispone de seis scripts que usan la herramienta minicom, los scripts son:

- scriptBackupConfiguration.sh

Este script que permite realizar copiar de la configuración actual del conmutador.

Este script requiere que el equipo y el conmutador esté conectado por un cable Ethernet además de la conexión serie, esto se debe a una limitación por la cual no se puede realizar la descarga del fichero de configuración mediante el protocolo XMODEM.

Para la ejecución de este script el conmutador puede encontrarse en modo recuperación.

Los parámetros para el script son:

-f backupConfiguracion	Indica la ruta absoluta donde se almacenará la configuración descargada. Valor por defecto backup_configuracion.
-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudRescue	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 115200.
-r	Indica si se inicia el proceso desde modo recuperación.
-p contraseña_manager	Indica la contraseña del administrador del conmutador.
-d directorioServidorTftp	Indica la ruta del directorio del servidor TFTP. Valor por defecto el configurado en la variable TFTP_DIRECTOTY de /var/default/tftpd-hpa o /var/lib/tftpboot/.
-P puertoSwitch	Indica el puerto ethernet del conmutador que se usará para conectar mediante TFTP. Valor por defecto 1.
-i interfazRedEthernet	Indica la interfaz del equipo que se usará para la conexión mediante Ethernet con el conmutador. Valor por defecto primer interfaz de red en orden alfabético.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptEnable.template con la contraseña de administrador indicada y un archivo temporal a partir de MinicomScripts/scriptBackupConfiguration.template con la configuración necesaria para realizar la conexión mediante TFTP para realizar la descarga de la configuración startup-config que se encuentre en el conmutador.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Iniciar el conmutador si este se indicó que se encontraba en modo recuperación.
3. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
4. Asociar una dirección IP temporal a la interfaz indicada en el parámetro interfazRedEthernet.
5. Crear una VLAN temporal con una dirección IP temporal en el conmutador para realizar la descarga mediante TFTP.
6. Asociar el puerto indicado en el parámetro puertoSwitch a la VLAN temporal.
7. Descargar la configuración startup-config actual.
8. Copiar el fichero de configuración donde se indicó con el parámetro backupConfiguracion.
9. Eliminar la dirección IP temporal del equipo.
10. Eliminar los archivos temporales.

El fichero de configuración se almacenará donde se indicó con el parámetro backupConfiguracion y su nombre contendrá la fecha de en la que se realizó.

- scriptLoadConfiguration.sh

Este script que permite cargar una configuración al conmutador,

Los parámetros para el script son:

-f backupConfiguracion	Indica el fichero de configuración que se usará. El valor por defecto es el fichero con nombre backup_configuracion que se encuentre en el mismo directorio que el script.
-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudRescue	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 115200.
-p contraseña_manager	Indica la contraseña del administrador del conmutador.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptEnable.template con la contraseña de administrador indicada y un archivo temporal a partir de MinicomScripts/scriptLoadConfiguration.template con el nombre del fichero de configuración a cargar.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
3. Mostrar por pantalla el nombre del fichero de configuración que está en uso actualmente.
4. Solicitar introducir la ubicación del fichero de configuración dentro del flash, para ello mostrará los posibles lugares.

5. Iniciar el modo de recuperación del conmutador.
6. Cargar el fichero de configuración.
7. Volver al modo normal del conmutador para cargar la configuración
8. Eliminar los archivos temporales.

- scriptUpgrade.sh

Este script que permite actualizar el firmware del conmutador.

Los parámetros para el script son:

-f ficheroFirmware	Indica el fichero de fi que se usará firmware. El fichero debe tener el formato bin. El valor por defecto es el fichero con nombre firmware.bin que se encuentre en el mismo directorio que el script.
-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudRescue	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 115200.
-p contraseña_manager	Indica la contraseña del administrador del conmutador.
-d	Indica si se eliminará las contraseñas y configuración en el proceso.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptEnable.template con la contraseña de administrador indicada y un archivo temporal a partir de MinicomScripts/scriptEndUploadFirmware.template con el nombre del fichero firmware a cargar.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
3. Mostrar por pantalla la ruta del fichero firmware que está en uso actualmente.
4. Solicitar introducir la ubicación del fichero firmware dentro del flash, para ello mostrará los posibles lugares.
5. Crear el fichero temporal a partir de MinicomScripts/scriptStartUploadFirmware.template con la ruta del firmware actual y con el nuevo firmware.
6. Iniciar el modo de recuperación del conmutador.
7. Si se indicó que se quería realizar el borrado de configuración y contraseña, se creará un archivo temporal a partir de MinicomScripts/ scriptRemovePasswordAndConfig.template con la ubicación del fichero de configuración que se le pedirá introducir y tras esto hará una copia del fichero en la ubicación flash:config_last_delete_password.txt(por si se quisiera recuperar) y lo eliminará de la ubicación actual.
8. Cargar el fichero de firmware.
9. Configurar el arranque para que haga uso del nuevo firmware.
10. Volver al modo normal del conmutador con el nuevo firmware.
11. Eliminar los archivos temporales.

- scriptDisableDHCP.sh

Este script deshabilitará el servicio de DHCP del conmutador.

Los parámetros para el script son:

-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudRescue	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 115200.
-p contraseña_manager	Indica la contraseña del administrador del conmutador.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptEnable.template con la contraseña de administrador indicada.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
3. Deshabilitar el servicio de DHCP en el conmutador.
4. Eliminar los archivos temporales.

- scriptEnableDHCP.sh

Este script habilitará el servicio de DHCP del conmutador.

Los parámetros para el script son:

-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudRescue	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 115200.
-p contraseña_manager	Indica la contraseña del administrador del conmutador.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptEnable.template con la contraseña de administrador indicada.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
3. Habilitar el servicio de DHCP en el conmutador.
4. Eliminar los archivos temporales.

- scriptEnableManagementWeb.sh

Este habilitará la web de gestión del conmutador.

Los parámetros para el script son:

-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudRescue	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 115200.
-p contraseña_manager	Indica la contraseña del administrador del conmutador.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptEnable.template con la contraseña de administrador indicada.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
3. Habilitar la web de gestión del conmutador.
4. Eliminar los archivos temporales.

5.2.2.2 Conmutador HP 2620-25 o 2650-25

Se dispone de seis scripts que usan la herramienta minicom, los scripts son:

- scriptBackupConfiguration.sh

Este que permite realizar copiar de la configuración actual del conmutador.

Los parámetros para el script son:

-f backupConfiguracion	Indica la ruta absoluta donde se almacenará la configuración descargada. Valor por defecto backup_configuracion.
-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudIncial	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 9600.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptBackupConfiguration.template el usuario y contraseña del administrador indicado.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
3. Descargar la configuración startup-config actual.

4. Eliminar los archivos temporales.

El fichero de configuración se almacenará donde se indicó con el parámetro backupConfiguracion.

- scriptLoadConfiguration.sh

Este script que permite cargar una configuración al conmutador,

Los parámetros para el script son:

-f backupConfiguracion	Indica el fichero de configuración que se usará. El valor por defecto es el fichero con nombre backup_configuracion que se encuentre en el mismo directorio que el script.
-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudIncial	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 9600.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptEnable.template con el usuario y contraseña del administrador indicado.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
3. Cargar el fichero de configuración.
4. Reiniciar del conmutador para cargar la configuración.
5. Eliminar los archivos temporales.

- scriptUpgrade.sh

Este script que permite actualizar el firmware del conmutador.

Los parámetros para el script son:

-f ficheroFirmware	Indica el fichero de fi que se usará firmware. El valor por defecto es el fichero con nombre firware.bin que se encuentre en el mismo directorio que el script.
-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudIncial	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 9600.
-r	Indica que se inicia desde modo recuperación por lo cual no sería necesario ni usuario_manager ni contrasena_manager
-u usuario_manager	Indica el usuario del administrador del conmutador.

-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptReset.template con el usuario y contraseña del administrador indicado.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Si no se indicó que se indicaba en modo recuperación, se habilitar los privilegios para realizar cambios en la configuración del conmutador y se inicará el modo recuperación.
3. Se cambiará la velocidad de baud a 115200 bauds.
4. Mostrar por pantalla la ruta del fichero firmware que está en uso actualmente.
5. Cargar el fichero de firmware.
6. Configurar el arranque para que haga uso del nuevo firmware.
7. Volver al modo normal del conmutador con el nuevo firmware.
8. Eliminar los archivos temporales.

- scriptDisableDHCP.sh

Este script deshabilitará el servicio de DHCP del conmutador.

Los parámetros para el script son:

-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudIncial	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 9600.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptDisaableDHCP.template con la contraseña de administrador indicada.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
3. Deshabilitar el servicio de DHCP en el conmutador.
4. Eliminar los archivos temporales.

- scriptEnableDHCP.sh

Este script habilitará el servicio de DHCP del conmutador.

Los parámetros para el script son:

-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudIncial	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 9600.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptEnableDHCP.template el usuario y contraseña del administrador indicado.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
3. Habilitar el servicio de DHCP en el conmutador.
4. Eliminar los archivos temporales.

- scriptEnableManagementWeb.sh

Este habilitará la web de gestión del conmutador.

Los parámetros para el script son:

-t tty_serie	Indica el dispositivo TTY donde está ubicado la conexión por puerto serie al conmutador. Valor por defecto /dev/ttyS0.
-b baudIncial	Indica los bauds con los cuales se ha iniciado el conmutador. Valor por defecto 9600.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Una vez ejecutado creará un archivo temporal a partir de MinicomScripts/scriptEnableManagementWeb.template con el usuario y contraseña del administrador indicado.

Con los archivos temporal comenzará el proceso cuyo paso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar los privilegios para realizar cambios en la configuración del conmutador.
3. Habilitar la web de gestión del conmutador.
4. Eliminar los archivos temporales.

5.2.3 Script para la gestión de conmutadores basado en el uso de SSH

serie con el conmutador mediante la ssh en sistema operativos basado en Linux. Estos deberán ser ejecutado superusuario.

Para el uso de los scripts tiene las siguientes dependencias:

- ssh, herramienta con la que se realizará la conexión y que ejecutará los scripts con los comandos especifico de cada conmutador.
- scp, herramienta con la que se realizará las transferencias de ficheros.
- expect[120], herramienta encargada de ejecutar los scripts sobre ssh y que permitirá la autentificación de ssh sin intervención.

Tambien será necesario conocer los siguientes datos:

- Usuario del administrador.
- Contraseña del administrador.
- Dirección IP del conmutador y
- Puerto SSH del conmutador.

Para estos scripts el funcionamiento es igual para todos los modelos de los conmutadores.

Para la ejecución de los scripts es necesario que tanto ssh como scp este activo en los conmutadores, el modo de hacerlo se detallará en el ANEXO L: Scripts de gestión de conmutadores.

Se dispone de seis scripts que usan la herramienta ssh, los scripts son:

- scriptBackupConfiguration.sh

Este que permite realizar copiar de la configuración actual del conmutador.

Los parámetros para el script son:

-f backupConfiguracion	Indica la ruta absoluta donde se almacenará la configuración descargada. Valor por defecto backup_configuracion.
-H direccion_switch	Indica la dirección del conmutador.
-P puerto_ssh_switch	Indica el puerto ssh configurado en el conmutador.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Los pasos que realizará el proceso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Descargar la configuración startup-config actual mediande scp.

- scriptLoadConfiguration.sh

Este script que permite cargar una configuración al conmutador,

Los parámetros para el script son:

-f backupConfiguracion	Indica el fichero de configuración que se usará. El valor por defecto es el fichero con nombre backup_configuracion que se encuentre en el mismo directorio que el script.
-H direccion_switch	Indica la dirección del conmutador.
-P puerto_ssh_switch	Indica el puerto ssh configurado en el conmutador.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Los pasos que realizará el proceso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Cargar la configuración en el fichero startup-config actual mediande scp.
3. Cargar la configuración en el fichero running-config actual mediande scp

- scriptUpgrade.sh

Este script que permite actualizar el firmware del conmutador.

Los parámetros para el script son:

-f ficheroFirmware	Indica el fichero de fi que se usará firmware. El valor por defecto es el fichero con nombre firmware.bin que se encuentre en el mismo directorio que el script.
-H direccion_switch	Indica la dirección del conmutador.
-P puerto_ssh_switch	Indica el puerto ssh configurado en el conmutador.
-i imagen_destiono	Indica la imagen destino del conmutador solo puede ser primary ó secondary. Solo es posible para HP y el valor por defecto es primary.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Los pasos que realizará el proceso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Cargar el nuevo firmware mediante scp
3. En caso de no tener suficiente espacio eliminar la imagen anterior para el conmutador de CISCO.
4. Reiniciar el conmutador con el nuevo firmware.

- scriptDisableDHCP.sh

Este script deshabilitará el servicio de DHCP del conmutador.

Los parámetros para el script son:

- H direccion_switch	Indica la dirección del conmutador.
-P puerto_ssh_switch	Indica el puerto ssh configurado en el conmutador.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Los pasos que realizará el proceso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Deshabilitar el servicio DHCP.

- scriptEnableDHCP.sh

Este script habilitará el servicio de DHCP del conmutador.

Los parámetros para el script son:

- H direccion_switch	Indica la dirección del conmutador.
-P puerto_ssh_switch	Indica el puerto ssh configurado en el conmutador.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Los pasos que realizará el proceso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar el servicio de DHCP en el conmutador.

- scriptEnableManagementWeb.sh

Este habilitará la web de gestión del conmutador.

Los parámetros para el script son:

- H direccion_switch	Indica la dirección del conmutador.
-P puerto_ssh_switch	Indica el puerto ssh configurado en el conmutador.
-u usuario_manager	Indica el usuario del administrador del conmutador.
-p contrasena_manager	Indica la contraseña del administrador del conmutador.

Los pasos que realizará el proceso son:

1. Comprobar que todos los parámetros fueron introducidos o tiene valor por defecto y las dependencias necesarias.
2. Habilitar la web de gestión del conmutador.

5.2.4 Líneas de continuación

Una posible línea continuación puede ser seguir añadiendo nuevos procesos realzado mediante script o añadir más modelos.

Otra posible línea de continuación puede crear una web que lance estos scripts y te permita realizar todas las operaciones necesarias mediante ella.

5.3 Plan de contingencia antes fallo en la red

En esta sección se detallará una serie de procesos a realizar para poder recuperar la conectividad de la red interna del laboratorio en caso de ocurrir errores en la misma.

Para poder realizar este proceso será necesario la configuración de un conmutador de respaldo que será usado en los procesos.

5.3.1 Configuración del conmutador de respaldo

A continuación, se detallará la configuración que se realizará en el conmutador que se tendrá como respaldo y que normalmente se mantendrá fuera de la red.

Antes de comenzar se recuerda que la subred A esta compuesta por las direcciones IP dentro del rango 172.16.17.0/25 y la subred B está compuesta por las direcciones IP dentro del rango 172.16.17.128/25. Y el servidor ait08.us.es tiene asignada las direcciones IP 172.16.17.126 para la subred A y 172.16.17.254 para la subred B y es el encargado de realizar el enrutamiento entre subredes y con el exterior

Se configurará las siguientes VLANs:

- VLAN 2 con la dirección IP asignada 172.16.17.125 y con los puertos asignados 1 y 2.
- VLAN 3 con la dirección IP asignada 172.16.17.253 y con los puertos asignados 3 y 4.
- VLAN 4 con la dirección IP asignada 172.16.17.126 y con los puertos asignados 5 y 6.
- VLAN 5 con la dirección IP asignada 172.16.17.254 y con los puertos asignados 7 y 8.
- VLAN 6 con la dirección IP asignado mediante DHCP y con los puertos asignados 21 y 22.
- VLAN 7 con la con la dirección IP asignada en el rango de ait.us.es (193.147.162.163) y con los puertos asignados 23 y 24.

Se configurará el enrutamiento IP en el conmutador.
Se configurará el servidor DHCP con los mismos parámetros que se encuentra actualmente en el servidor ait08.us.es, para ello se proporcionará el script mapeoDHCPcompleto.pl junto a la memoria que realizará la lectura del fichero dhcpd.conf y generará la configuración DHCP necesaria para modelos compatibles con HP 2620-25, HP 2650-25 o Cisco 3750G-24TS.
Se configurará las siguientes rutas por defecto:

- 0.0.0.0/0 hacia la dirección IP de la puerta enlace predeterminada de ait.us.es con métrica 1
- 0.0.0.0/0 hacia la dirección IP de la puerta enlace predeterminada para la subred del AP con métrica 2

En los puertos del 1 al 8 no se conectará ningún cable de Ethernet
En el puerto 21 o 22 se conectará con un punto de acceso que estará conectado a la red Eduroam y permitirá la comunicación del laboratorio con el exterior a través de esta conexión.
En el puerto 23 o 24 se conectará con la red ait.us.es y permitirá la comunicación del laboratorio con el exterior a través de esta conexión cableada.

El script mapeoDHCPcompleto.pl junto a la invocación con un solo argumento que sea la ruta de un fichero de configuración de un servidor DHCP valido dará como resultado un fichero, cuyo nombre se especifica en la función nombreFicheroSalida, contendrá los comandos o parte de la configuración (en función de si en la función modoFicheroSalida se especificó comando o ficheroConfiguracion).
Tambien se generará un fichero con los rangos configurado para cada subred configurada por si se desea crear otra VLAN que ofrezca IP dinámicas en un rango distinto del ya configurado El nombre de este fichero se especificará en la función nombreFicheroSalidaRangos.

Junto a la memoria se entregará los ficheros indicando los comandos necesarios para realizar la configuración indica para modelos compatibles con HP 2620-25, HP 2650-25 o Cisco 3750G-24TS.

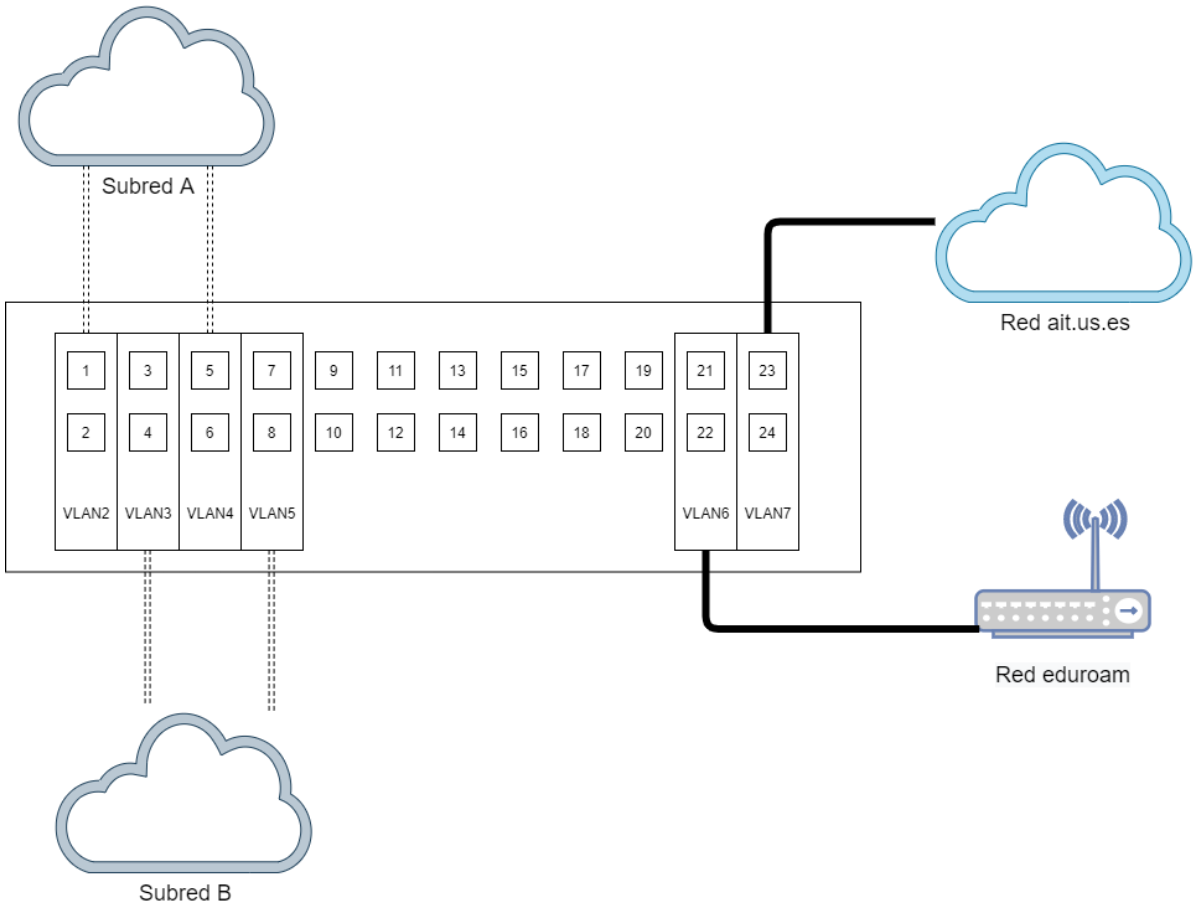

Ilustración 5-2 Esquema de conexión del conmutador de respaldo

5.3.2 Fallo de enrutamiento

Los pasos que se deben realizar en caso de que falle el enrutamiento son los siguientes:

1. Conectar el cable que la subred A en el puerto 1 o 2.
2. Conectar el cable que la subred B en el puerto 3 o 4.
3. Desactivar el servicio del DHCP en el conmutador.
4. Ejecutar en los clientes el comando para cambiar de puerta de enlace tras recibir la configuración por DHCP de los servidores.

Para cliente en la subred A

- En Linux

```
ip route add default via 172.16.17.125 dev internazEnLaSubredA
route add default gw 172.16.17.125 internazEnLaSubredA
```

- En Windows

```
route add 0.0.0.0 mask 0.0.0.0 172.16.17.125 metric 30 if
numero_intefaz_SubredA
```

Para cliente en la subred B

- En Linux

```
ip route add default via 172.16.17.253 dev internazEnLaSubredB
route add default gw 172.16.17.253 internazEnLaSubredB
```

- En Windows

```
route add 0.0.0.0 mask 0.0.0.0 172.16.17.253 metric 30 if
numero_intefaz_SubredB
```

Para ver el número de la interfaz en Windows de salida podemos ejecutar el comando

```
netsh interface ipv4 show interfaces
```

```
C:\Users\Administrador>netsh interface ipv4 show interfaces

Índ     Mét         MTU          Estado                Nombre
---  ----------  ----------  ------------  ---------------------------
  1          75  4294967295  connected     Loopback Pseudo-Interface 1
 24          55        1500  connected     Wi-Fi
 22           5        1500  disconnected  Ethernet
 18          25        1500  disconnected  Conexión de área local* 1
 14          25        1500  disconnected  Conexión de área local* 2
  7          35        1500  connected     VMware Network Adapter VMnet1
 16          35        1500  connected     VMware Network Adapter VMnet8
 20          25        1500  connected     VirtualBox Host-Only Network
  5          15        1500  connected     vEthernet (Modificador pre)
 27          10        1500  disconnected  Ethernet 3
 11          25       65536  connected     Npcap Loopback Adapter
 19          25        1500  connected     Ethernet 4
 10          65        1500  disconnected  Conexión de red Bluetooth
```

5.3.3 Fallo tanto enrutamiento como el servicio DHCP

Los pasos que se deben realizar en caso de que falle tanto el enrutamiento como el servicio DHCP son los siguientes:

1. Conectar el cable que la subred A en el puerto 5 o 6.
2. Conectar el cable con la subred B en el puerto 7 o 8.
3. Activar el servicio del DHCP en el conmutador.
4. Ejecutar en los clientes el comando para solicitar de nuevo direcciones IP mediante DHCP.

En Linux

```
dhclient -v -r internazEnLaSubred
```

En Windows

```
ipconfig /release internazEnLaSubred
ipconfig /renew internazEnLaSubred
```

5.3.4 Líneas de continuación

Una línea de continuación es la configuración NAT para realizar el enrutamiento entre las subredes A y B y la red externa cableada directamente, esto queda pendiente debido a que los modelos de los conmutadores disponible en el laboratorio no disponen de esta funcionalidad.

Otra línea de continuación es la configuración de un punto de acceso que se encargará de conectarse a la red de Eduroam, ofrecer una subred donde se conectará el conmutador de respaldo y realizar NAT entre la subred y Eduroam. Para ello se hará uso de OpenWrt en un punto de acceso Linksys WRT3200ACM-EU que se adquirido recientemente para su uso en el Laboratorio de Telemática.

6 CONCLUSIONES

Con este trabajo fin de grado se han abordado una serie de mejoras que permitirá en un futuro que el laboratorio de Telemática se más robusto ante fallo y se pueda recuperar antes estos fallos con mayor facilidad.

Entre las mejoras que se han planteado también están aquellas destinadas a facilitar la administración y mantenimiento del laboratorio, como puede ser la inclusión de nuevas funcionalidades en Opengnsys o los scripts para la gestión de maquina virtuales o conmutadores.

Por último se han abordado algunas soluciones que permitirá el uso de los equipos de laboratorio de forma remota que ser de gran utilidad en situaciones en los que no se accesible el laboratorio como ha ocurrido recientemente.

Todas estas mejoras abordadas son un únicamente paso en todos los posibles cambios que se pueden realizar y como se ha indicado en cada sección todas estas soluciones posee líneas de continuación que pueden ser abordadas en futuro.

ANEXO A: CLONACIÓN DE RAID MEDIANTE SOFTWARE

En este anexo se detallarán el concepto de superbloque cuando se habla de RAID software, como cambian los UUID de las particiones o dispositivos en el proceso de creación de RAID y como se relacionan con el RAID y por último se mostrarán los scripts necesarios para realizar la clonación y restauración automáticamente del escenario propuesto en la sección Clonación de RAID mediante software.

Superbloque en RAID software

Un superbloque o superblock es un conjunto de sectores de los dispositivos de almacenamiento que componen el RAID. En él se almacena toda la información relativa al RAID, como puede ser el tipo de RAID que forma, el UUID del mismo o los dispositivos que lo componen. Existen actualmente cuatro versiones de superbloque. La mayor diferencia entre estas versiones es la posición del dispositivo de almacenamiento donde se localiza el superbloque.

Las diferentes versiones son:

- v0.9 Se almacena al final del dispositivo de almacenamiento.
- v1.0 Se almacena al final del dispositivo de almacenamiento.
- v1.1 Se almacena al principio del dispositivo de almacenamiento.
- v1.2 Se almacena a 4 Kb desde el principio del dispositivo de almacenamiento.

La versión de superbloque se puede seleccionar al crear el RAID mediante la opción --metadata .

Entre la versión 0.9 y las posteriores existen dos diferencias: hubo un aumento en el número de dispositivos que podían componer el RAID, de 28 a 384, y el límite de tamaño de los discos soportado pasó de 2TB a 4TB.

Se recomienda el uso de las versiones 1.1 y 1.2 debido a que algunos sistemas de auto montaje podrían detectar sistemas de ficheros que se encuentran dentro de los dispositivos. Si esto pasara, estos sistemas podrían montar estos archivos en lugar del RAID. Además, si se desea clonar en dispositivos de mayor capacidad, se tendrían problemas para expandir la partición tras realizar la clonación.

La ubicación del superbloque será la misma a partir del inicio del dispositivo o de la partición para todos los que forman el RAID, es decir, comparten el sector desde el que comienza el superbloque empezando desde el inicio del dispositivo o partición.

El tamaño del superbloque depende de la versión:

- Para la versión 0.9 el tamaño es de 4KB.
- Para la versión 1.0 y posteriores es de 256B más 2B por cada dispositivo que forme parte del RAID.

Una vez creado el RAID se dispondrá de un dispositivo sobre el que se creará el sistema de ficheros. El superbloque o superblock quedará fuera de dicho sistema de ficheros. Esto provoca una pérdida de capacidad efectiva de almacenamiento.

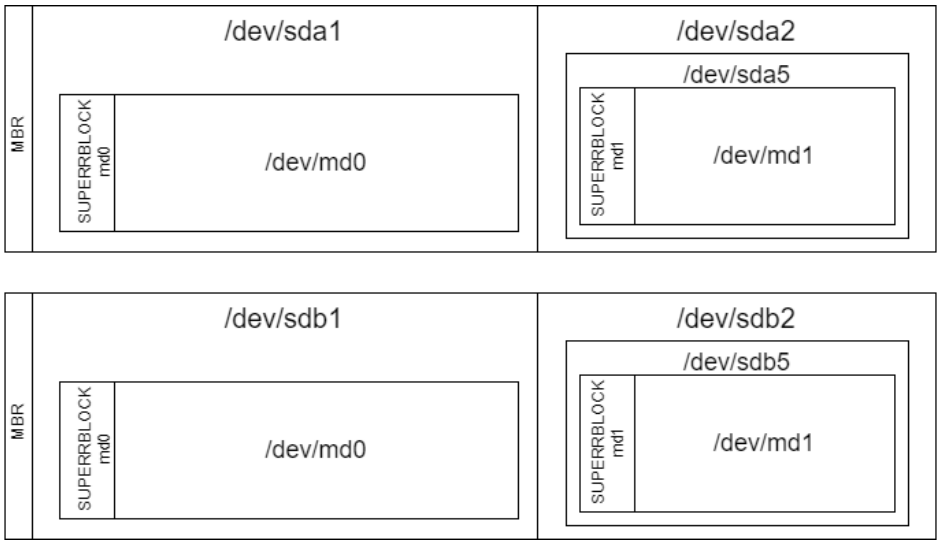

Ilustración A-0-1 Diagrama ejemplo de ubicación del superblock version 1.1

UUIDs de dispositivos perteneciente a un RAID software

En este aparatado se detalla como el sistema operativo identifica a los dispositivos que componen un determinado RAID. Este proceso se produce como resultado de los cambios de UUID de los dispositivos cuando se incluyen en un RAID.

En los diagramas que se muestran se puede observar el RAID /dev/md0 compuesto por los dispositivos /dev/sda1 y /dev/sdb1 con sus respetivos UUID, además de donde se ubican estos en el escenario propuesto. También enseña el tipo de formato que posee cada uno.

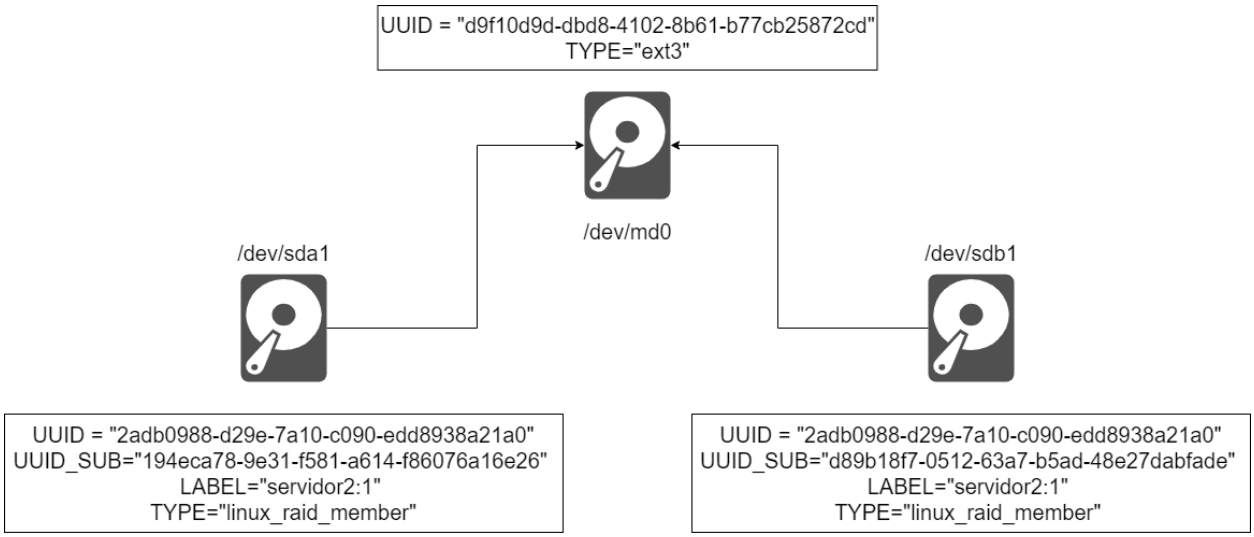

Ilustración A-0-2 Esquema de UUID en un RAID software

Como se puede observar en el diagrama, una vez que se crea el RAID se producen los siguientes cambios en los dispositivos o particiones:

- Los UUIDs de los dispositivos pasan a ser uno común. Este dato lo identifica a cada uno de ellos como parte de un RAID. Este sería el mismo UUID que identifica el dispositivo nuevo RAID.
- El antiguo UUID de cada dispositivo o partición se guardaría en otro dato que sería el identificador UUID_SUB.
- A su vez, cuando se formatee con un sistema de archivo el RAID o cuando se particione, se le generará

otro UUID distinto al anterior.

- Por último, los dispositivos que componen el RAID pasarán a tener como etiqueta el nombre que se le asigne al RAID.

A continuación, se mostrarán una serie de capturas obtenidas con la herramienta blkid y mdadm, donde se puede ver todos los datos que se han comentado:

```
root@lt-servidor2:~# blkid /dev/sd[ab]1 /dev/md0
/dev/sda1: UUID="2adb0988-d29e-7a10-c090-edd8938a21a0" UUID_SUB="d89b18f7-0512-63a7-b5ad-48e2
7dabfade" LABEL="servidor2:1" TYPE="linux_raid_member" PARTUUID="00013f01-01"
/dev/sdb1: UUID="2adb0988-d29e-7a10-c090-edd8938a21a0" UUID_SUB="194eca78-9e31-f581-a614-f860
76a16e26" LABEL="servidor2:1" TYPE="linux_raid_member" PARTUUID="2bbf801f-01"
/dev/md0: UUID="d9f10d9d-dbd8-4102-8b61-b77cb25872cd" TYPE="ext3"
```

Ilustración A-0-3 Resultado de consulta UUID de /dev/sda, /dev/sdb y /dev/md0

En esta captura podemos ver los UUID tanto de los dispositivos que componen el RAID como el UUID de la partición única que se realizó sobre el RAID.

```
root@lt-servidor2:~# mdadm -E /dev/sd[ab]1
/dev/sda1:
          Magic : a92b4efc
        Version : 1.2
    Feature Map : 0x0
     Array UUID : 2adb0988:d29e7a10:c090edd8:938a21a0
           Name : servidor2:1
  Creation Time : Sun May 31 16:31:18 2020
     Raid Level : raid1
   Raid Devices : 2

 Avail Dev Size : 39028736 (18.61 GiB 19.98 GB)
     Array Size : 19514368 (18.61 GiB 19.98 GB)
    Data Offset : 32768 sectors
   Super Offset : 8 sectors
   Unused Space : before=32680 sectors, after=0 sectors
          State : clean
    Device UUID : d89b18f7:051263a7:b5ad48e2:7dabfade

    Update Time : Sat Feb 16 08:07:35 2019
  Bad Block Log : 512 entries available at offset 72 sectors
       Checksum : c37cf704 - correct
         Events : 84


   Device Role : Active device 0
   Array State : AA ('A' == active, '.' == missing, 'R' == replacing)
/dev/sdb1:
          Magic : a92b4efc
        Version : 1.2
    Feature Map : 0x0
     Array UUID : 2adb0988:d29e7a10:c090edd8:938a21a0
           Name : servidor2:1
  Creation Time : Sun May 31 16:31:18 2020
     Raid Level : raid1
   Raid Devices : 2

 Avail Dev Size : 39028736 (18.61 GiB 19.98 GB)
     Array Size : 19514368 (18.61 GiB 19.98 GB)
    Data Offset : 32768 sectors
   Super Offset : 8 sectors
   Unused Space : before=32680 sectors, after=0 sectors
          State : clean
    Device UUID : 194eca78:9e31f581:a614f860:76a16e26

    Update Time : Sat Feb 16 08:07:35 2019
```

Ilustración A-0-4 Resultado consulta mdadm -E /dev/sda /dev/sdb

En esta captura se puede observar el resultado del comando:

```
mdadm -E /dev/sd[ab]1
```

Presenta el campo Device UUID que muestra los UUIDs de los dispositivos que componen el RAID (corresponde con el campo SUB_UUID de la herramienta blkid), el campo Array UUID que muestra el UUID del RAID al que pertenece y el nombre del RAID que tienen ambos dispositivos como etiqueta en el campo NAME.

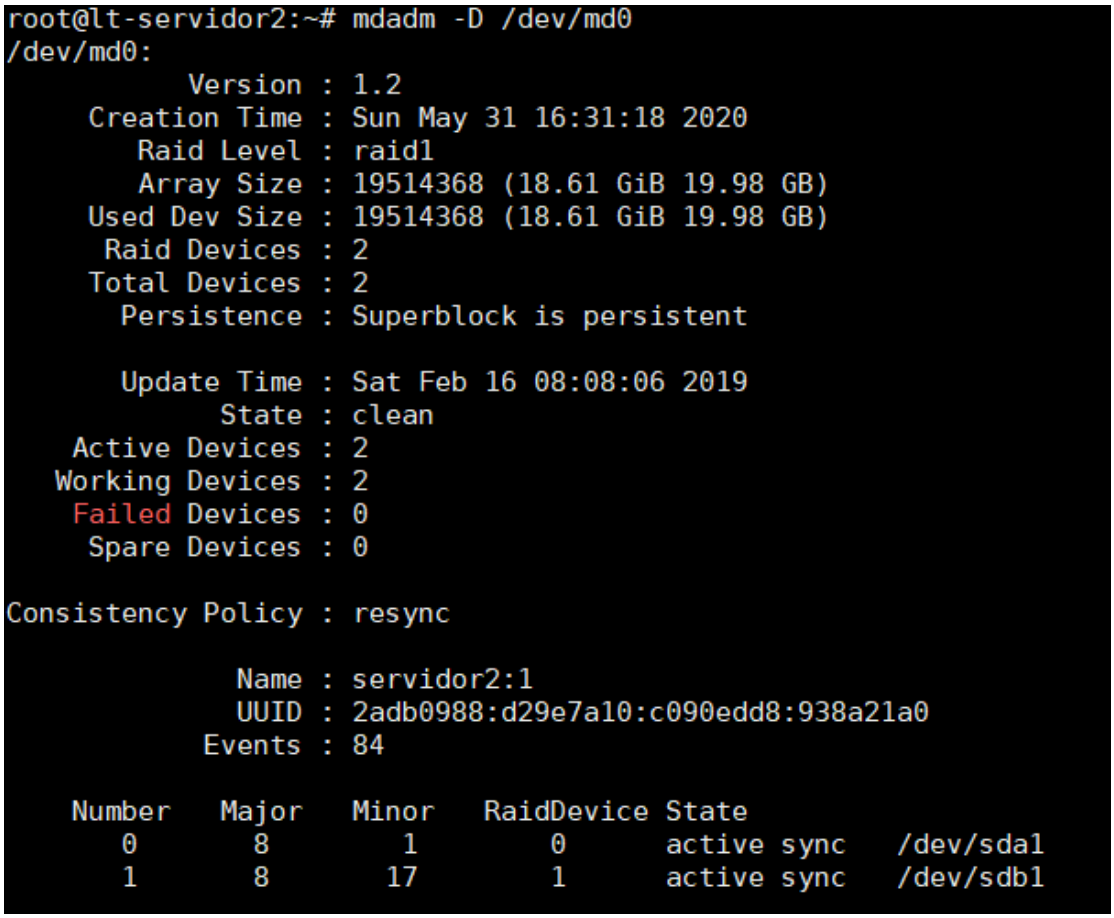

Ilustración A-0-5 Resultado consulta mdadm -D /dev/md0

Por último, en esta captura podemos observar el resultado del comando

```
mdadm -D /dev/md0
```

Presenta los dispositivos que componen el RAID al final de la captura, el nombre de este en el campo Name y su UUID en el campo con el mismo nombre.

En la siguiente tabla se hace un resumen de los UUIDs que se han detallado anteriormente.

Tabla 3 Tipos de UUID en RAID software

Tipo de UUID	Significado	Manera de comprobar
UUID del array	Este UUID identifica al RAID en el sistema. Este es compartido por todos los dispositivos de un array	`blkid /dev/sdXY` y se muestra en el campo UUID `mdadm -E /dev/sdXY` y se muestra en el campo Array UUID
UUID de dispositivo del array	Este UUID identifica de manera única en el sistema a un dispositivo perteneciente a un RAID	`blkid /dev/sdXY` y se muestra en el campo UUID_SUB `mdadm -E /dev/sdXY` y se muestra en el campo Device UUID
UUID partición del array	Este UUID identifica una partición creada en el RAID y a la que se le ha dado un formato	`blkid /dev/mdX` y se muestra en el campo UUID

Script de automatización para la clonación y restauración de RAID software

Se tomará como escenario de partida el detallado en el apartado Clonación de RAID mediante software.

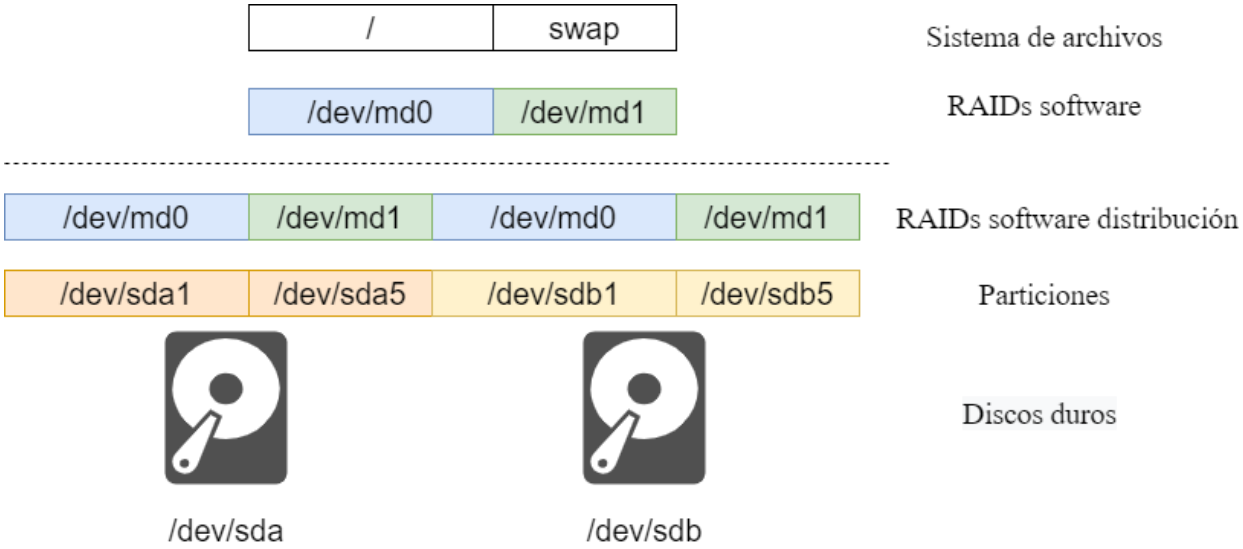

Ilustración A-0-6 Escenario propuesto para clonación RAID software

Script el proceso de clonación

El siguiente script permite la clonación completa del escenario.

En el script se realizan los siguientes pasos:

1. Parar y recrear los RAIDs, para asegurarnos que se nombran como md0 y md1.
2. Clonar los datos en el RAID md0.
3. Copiar el MBR de los discos.

4. Copiar tabla de particiones de los discos.
5. Buscar la ubicación de los superbloques.
6. Buscar el tamaño de los sectores.
7. Clonar los superbloques.
8. Guardar UUID y la etiqueta de la partición swap en el RAID md1.

```
 #!/bin/sh
 set -e
 echo "Comienza el proceso de clonacion"
 echo "Paramos los RAIDs y los recreamos correctamente"
 mdadm -S --scan
 mdadm -A --run --update=resync /dev/md0 /dev/sda1 /dev/sdb1
 mdadm -A --run --update=resync /dev/md1 /dev/sda5 /dev/sdb5

 echo "Copiando el MBR"
 dd if=/dev/sda of=backup_sda_MBR.dd bs=512 count=1
 dd if=/dev/sdb of=backup_sdb_MBR.dd bs=512 count=1

 echo " Copiando tabla de particiones"
 sfdisk -d /dev/sda > backup_sda.sfdisk
 sfdisk -d /dev/sdb > backup_sdb.sfdisk

 echo "Copiando supeblock RAIDs"

 echo "Buscando ubicacion de superblock"
 j=0
 for i in $(mdadm -E /dev/sda1 | grep "Super Offset"); do
  j=$((j+1))
  if [ "$j" = "4" ]; then
   echo "OFFSET_MD0=$i" > offset_superblock_md0
  fi
 done

 j=0
 for i in $(mdadm -E /dev/sda5 | grep "Super Offset"); do
  j=$((j+1))
  if [ "$j" = "4" ]; then
   echo "OFFSET_MD1=$i" > offset_superblock_md1
  fi
 done

 echo "Buscando tamanio de los sectores"
 echo "SECTOR_SIZE_SDA=$(cat /sys/block/sda/queue/hw_sector_size)" >
sector_size_sda
 echo "SECTOR_SIZE_SDB=$(cat /sys/block/sda/queue/hw_sector_size)" >
sector_size_sdb

 . sector_size_sda
 . sector_size_sdb

 echo "Clonando superblock del raid md0"
 . offset_superblock_md0
```

```
  dd if=/dev/sda1 of=sda1_superblock.dd count=1 bs="$SECTOR_SIZE_SDA"
skip="$OFFSET_MD0"
  dd if=/dev/sdb1 of=sdb1_superblock.dd count=1 bs="$SECTOR_SIZE_SDB"
skip="$OFFSET_MD0"

  echo "Clonando superblock del raid md1"
  . offset_superblock_md1
  dd if=/dev/sda5 of=sda5_superblock.dd count=1 bs="$SECTOR_SIZE_SDA"
skip="$OFFSET_MD1"
  dd if=/dev/sdb5 of=sdb5_superblock.dd count=1 bs="$SECTOR_SIZE_SDB"
skip="$OFFSET_MD1"

  echo "Guardando uuid y label de swap"
  for i in $(blkid /dev/md1); do
   case "$i" in
   UUID*)
    echo "$i" |sed "s/UUID/UUID_MD1/g" > uuid_md1 ;;
   LABEL*)
    echo "$i" | sed "s/LABEL/LABEL_MD1/g" > label_md1 ;;
   esac
  done
  echo "Terminado el proceso de clonacion correctamente"
```

Script para el proceso de restauración

El siguiente script permite la restauración completa del escenario.

En el script se realizan los siguientes pasos:

1. Restaurar el MBR de los discos.
2. Restaurar tabla de particiones de los discos.
3. Restaurar los superbloques.
4. Parar los RAIDs.
5. Recrear el RAID md1.
6. Recrear la partición swap en md1.
7. Recrear el RAID md0.
8. Restaurar los datos del RAID md0.
9. Reinstalar el GRUB.

```
  #!/bin/sh
  set -e
  echo "Comienza el proceso de restauracion"
  echo "Restaurando el MBR y tabla de particiones"
  echo "Restaurando MBR "
  dd if=backup_sda_MBR.dd of=/dev/sda bs=512 count=1
  dd if=backup_sdb_MBR.dd of=/dev/sdb bs=512 count=1

  echo "Restaurando tabla de particiones extendida"
  sfdisk /dev/sda < backup_sda.sfdisk
  sfdisk /dev/sdb < backup_sdb.sfdisk
```

```
  . sector_size_sda
  . sector_size_sdb

  echo "Restaurando superblock del raid md0"
  . offset_superblock_md0
  dd of=/dev/sda1 if=sda1_superblock.dd count=1 bs="$SECTOR_SIZE_SDA"
seek="$OFFSET_MD0"
  dd of=/dev/sdb1 if=sdb1_superblock.dd count=1 bs="$SECTOR_SIZE_SDB"
seek="$OFFSET_MD0"

  echo "Restaurando superblock del raid md1"
  . offset_superblock_md1
  dd of=/dev/sda5 if=sda5_superblock.dd count=1 bs="$SECTOR_SIZE_SDA"
seek="$OFFSET_MD1"
  dd of=/dev/sdb5 if=sdb5_superblock.dd count=1 bs="$SECTOR_SIZE_SDB"
seek="$OFFSET_MD1"

  echo "Parando raid si iniciaron automaticamente"
  mdadm -S --scan

  echo "Restaurando partición swap"
  mdadm -A --run --update=resync /dev/md1 /dev/sda5 /dev/sdb5
  . uuid_md1
  . label_md1
  mkswap -U "$UUID_MD1" -L "${LABEL_MD1:-"raid1Swap"}" /dev/md1

  echo "Restaurando datos RAID 1 con datos"
  mdadm -A --run --update=resync /dev/md0 /dev/sda1 /dev/sdb1
  fsarchiver -j2 -v restfs md0.fsa id=0,dest=/dev/md0

  echo "Reinstando el GRUB"
  mkdir md0_carpeta
  mount /dev/md0 md0_carpeta
  grub-install --boot-directory=md0_carpeta/boot/ /dev/sda
  grub-install --boot-directory=md0_carpeta/boot/ /dev/sdb

  echo "Terminado el proceso de restauracion correctamente"
```

ANEXO B: CLONACIÓN DE LVM

En este anexo se detallarán la estructura e información del fichero generado vgcfgbackup y se mostrarán los scripts necesarios para realizar la clonación y restauración automáticamente del escenario propuesto en la sección Clonación de LVM.

Estructura de fichero generado con vgcfgbackup.

En este apartado se explicará el contenido del fichero generado vgcfgbackup y que se usa como parte del proceso de restauración de un grupo de volúmenes.

Se parte del escenario propuesto en la sección Clonación de LVM. En el siguiente diagrama se presentarán los UUIDs de los dispositivos que son parte del escenario:

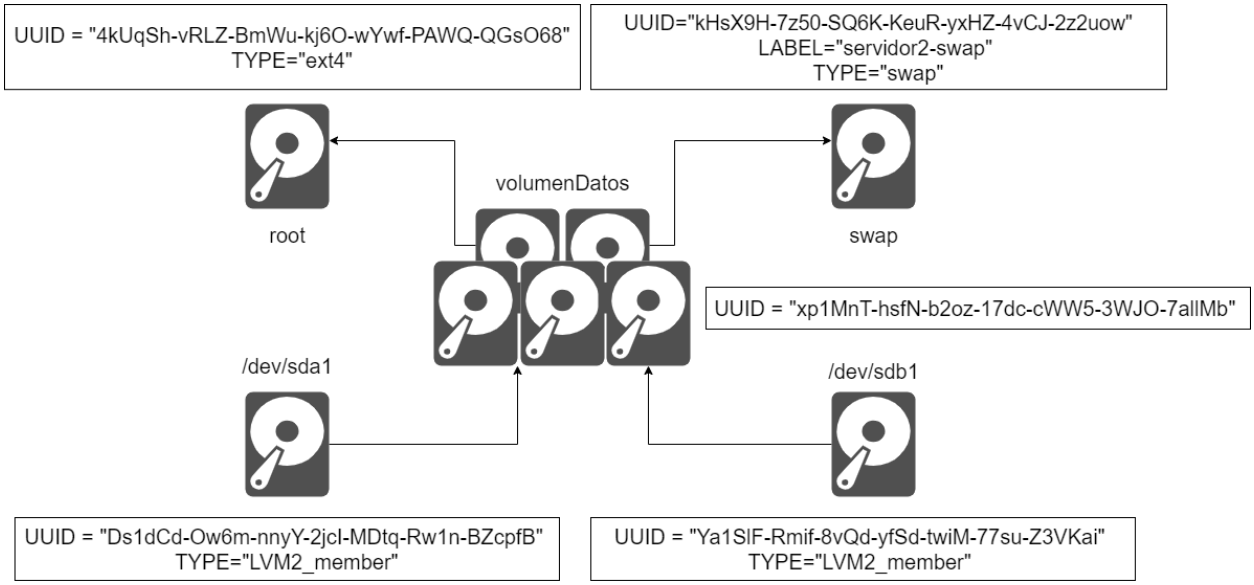

Ilustración B-0-1 Esquema de UUID en un LVM

A continuación, se mostrarán una serie de capturas obtenidas con la herramienta pvs[69] , lvs[69] y vgs[121], donde se puede observar todos los datos que se han comentado anteriormente.

```
[root@lt-servidor2 ~]# lvs -o lv_name,vg_name,lv_size,lv_uuid
  LV   VG           LSize   LV UUID
  root volumenDatos <18.50g 4kUqSh-vRLZ-BmWu-kj6O-wYwf-PAWQ-QGsO68
  swap volumenDatos   1.00g kHsX9H-7z50-SQ6K-KeuR-yxHZ-4vCJ-2z2uow
```

Ilustración B-0-2 Resultado de consulta UUID de volúmenes lógicos

```
[root@lt-servidor2 ~]# vgs -o vg_name,vg_size,vg_uuid
  VG           VSize   VG UUID
  volumenDatos <19.50g xp1MnT-hsfN-b2oz-17dc-cWW5-3WJO-7allMb
```

Ilustración B-0-3 Resultado de consulta UUID de grupo de volúmenes

```
[root@lt-servidor2 ~]# pvs -o pv_name,vg_name,pv_size,pv_uuid
  PV         VG           PSize PV UUID
  /dev/sda1  volumenDatos 9.50g Ds1dCd-Ow6m-nnyY-2jcI-MDtq-Rw1n-BZcpfB
  /dev/sdb1  volumenDatos 9.99g Ya1SlF-Rmif-8vQd-yfSd-twiM-77su-Z3VKai
```

Ilustración B-0-4 Resultado de consulta UUID de volúmenes físicos

A continuación, mostramos el fichero generado por la herramienta vgcfgbackup en el escenario propuesto:

```
  # Generated by LVM2 version 2.02.184(2) (2019-03-22): Wed Aug 19
21:46:37 2020

  contents = "Text Format Volume Group"
  version = 1

  description = "vgcfgbackup -f volumenDatosConfig volumenDatos"

  creation_host = "lt-servidor2" # Linux lt-servidor2 4.19.34-1-lts #1
SMP Sat Apr 6 19:41:19 CEST 2019 x86_64
  creation_time = 1597873597 # Wed Aug 19 21:46:37 2020

  volumenDatos {
   id = "xp1MnT-hsfN-b2oz-17dc-cWW5-3WJO-7allMb"
   seqno = 3
   format = "lvm2" # informational
   status = ["RESIZEABLE", "READ", "WRITE"]
   flags = []
   extent_size = 8192  # 4 Megabytes
   max_lv = 0
   max_pv = 0
   metadata_copies = 0

   physical_volumes {

    pv0 {
     id = "Ds1dCd-Ow6m-nnyY-2jcI-MDtq-Rw1n-BZcpfB"
     device = "/dev/sda1" # Hint only

     status = ["ALLOCATABLE"]
     flags = []
     dev_size = 19939328 # 9.50781 Gigabytes
     pe_start = 2048
     pe_count = 2433 # 9.50391 Gigabytes
    }
    pv1 {
     id = "Ya1SlF-Rmif-8vQd-yfSd-twiM-77su-Z3VKai"
     device = "/dev/sdb1" # Hint only

     status = ["ALLOCATABLE"]
     flags = []
     dev_size = 20963328 # 9.99609 Gigabytes
     pe_start = 2048
     pe_count = 2558 # 9.99219 Gigabytes
    }
   }
```

```
  logical_volumes {

   root {
    id = "4kUqSh-vRLZ-BmWu-kj6O-wYwf-PAWQ-QGsO68"
    status = ["READ", "WRITE", "VISIBLE"]
    flags = []
    creation_time = 1597785668 # 2020-08-18 21:21:08 +0000
    creation_host = "localhost"
    segment_count = 2

    segment1 {
     start_extent = 0
     extent_count = 2558 # 9.99219 Gigabytes

     type = "striped"
     stripe_count = 1 # linear

     stripes = [
      "pv1", 0
     ]
    }
    segment2 {
     start_extent = 2558
     extent_count = 2177 # 8.50391 Gigabytes

     type = "striped"
     stripe_count = 1 # linear

     stripes = [
      "pv0", 0
     ]
    }
   }

   swap {
    id = "kHsX9H-7z50-SQ6K-KeuR-yxHZ-4vCJ-2z2uow"
    status = ["READ", "WRITE", "VISIBLE"]
    flags = []
    creation_time = 1597785671 # 2020-08-18 21:21:11 +0000
    creation_host = "localhost"
    segment_count = 1

    segment1 {
     start_extent = 0
     extent_count = 256 # 1024 Megabytes

     type = "striped"
     stripe_count = 1 # linear

     stripes = [
      "pv0", 2177
     ]
    }
   }
  }
}
```

El primer fragmento proporciona información como el contenido del documento, la versión usada en el mismo, el comando con el cual se creó, el nombre del host donde se creó y cuándo se creó.

```
  # Generated by LVM2 version 2.02.184(2) (2019-03-22): Wed Aug 19
21:46:37 2020

  contents = "Text Format Volume Group"
  version = 1

  description = "vgcfgbackup -f volumenDatosConfig volumenDatos"

  creation_host = "lt-servidor2" # Linux lt-servidor2 4.19.34-1-lts #1
SMP Sat Apr 6 19:41:19 CEST 2019 x86_64
  creation_time = 1597873597 # Wed Aug 19 21:46:37 2020
```

Luego, se encuentra el inicio del objeto que define el grupo de volúmenes volumenDatos. En este comienzo se define el nombre del grupo de volúmenes.

```
  volumenDatos {
   id = "xp1MnT-hsfN-b2oz-17dc-cWW5-3WJO-7allMb"
   seqno = 3
   format = "lvm2" # informational
   status = ["RESIZEABLE", "READ", "WRITE"]
   flags = []
   extent_size = 8192  # 4 Megabytes
   max_lv = 0
   max_pv = 0
   metadata_copies = 0
```

Los campos más relevantes que se pueden observar:

- El campo id que identifica el UUID del grupo de volúmenes.
- El campo format que identifica la version de LVM de la que se está haciendo uso.
- El campo status que indica que el grupo de volúmenes puede ser redimensionado y usado para lectura y escritura.

En el siguiente fragmento se definen los volúmenes físicos que componen el grupo de volúmenes. En este ejemplo se puede ver que existen dos volúmenes físicos: pv0 que hace referencia a /dev/sda1 y pv1 que hace referencia a /dev/sdb1.

```
  physical_volumes {

    pv0 {
     id = "Ds1dCd-Ow6m-nnyY-2jcI-MDtq-Rw1n-BZcpfB"
     device = "/dev/sda1" # Hint only

     status = ["ALLOCATABLE"]
     flags = []
     dev_size = 19939328 # 9.50781 Gigabytes
     pe_start = 2048
```

```
    pe_count = 2433 # 9.50391 Gigabytes
   }

   pv1 {
    id = "Ya1SlF-Rmif-8vQd-yfSd-twiM-77su-Z3VKai"
    device = "/dev/sdb1" # Hint only

    status = ["ALLOCATABLE"]
    flags = []
    dev_size = 20963328 # 9.99609 Gigabytes
    pe_start = 2048
    pe_count = 2558 # 9.99219 Gigabytes
   }
  }
```

Los campos más relevantes que definen cada volumen físico son:

- El campo id que identifica el UUID del volumen físico.
- El campo device que identifica al dispositivo al que está asociado el volumen físico.
- El campo status que indica si está disponible para su uso.
- El campo dev_size que indica el tamaño que posee el volumen físico.

En el último fragmento se definen los volúmenes lógicos que se han creado en el grupo de volúmenes. En este ejemplo se puede ver que existen dos volúmenes lógicos, root y swap.

```
logical_volumes {

  root {
   id = "4kUqSh-vRLZ-BmWu-kj6O-wYwf-PAWQ-QGsO68"
   status = ["READ", "WRITE", "VISIBLE"]
   flags = []
   creation_time = 1597785668 # 2020-08-18 21:21:08 +0000
   creation_host = "localhost"
   segment_count = 2

   segment1 {
    start_extent = 0
    extent_count = 2558 # 9.99219 Gigabytes

    type = "striped"
    stripe_count = 1 # linear

    stripes = [
     "pv1", 0
    ]
   }
   segment2 {
    start_extent = 2558
    extent_count = 2177 # 8.50391 Gigabytes
```

```
      type = "striped"
      stripe_count = 1 # linear

      stripes = [
       "pv0", 0
      ]
     }
    }

    swap {
     id = "kHsX9H-7z50-SQ6K-KeuR-yxHZ-4vCJ-2z2uow"
     status = ["READ", "WRITE", "VISIBLE"]
     flags = []
     creation_time = 1597785671 # 2020-08-18 21:21:11 +0000
     creation_host = "localhost"
     segment_count = 1

     segment1 {
      start_extent = 0
      extent_count = 256 # 1024 Megabytes

      type = "striped"
      stripe_count = 1 # linear

      stripes = [
       "pv0", 2177
      ]
     }
    }
   }
```

Los campos más relevantes que definen cada volumen lógico son:

- El campo id que identifica el UUID del volumen lógico.
- El campo status que indica si está disponible para realizar lectura y escritura y si es visible para el usuario.
- El campo segment_count que es el número de segmento que compone al volumen lógico.
- El objeto segmentX que define uno de los segmentos que compone el volumen lógico. Con ello define su tamaño, el tipo de segmento y en qué volumen lógico se encuentra.

Script de automatización para la clonación y restauración de LVM

Se tomará como escenario de partida el detallado en el apartado Clonación de LVM

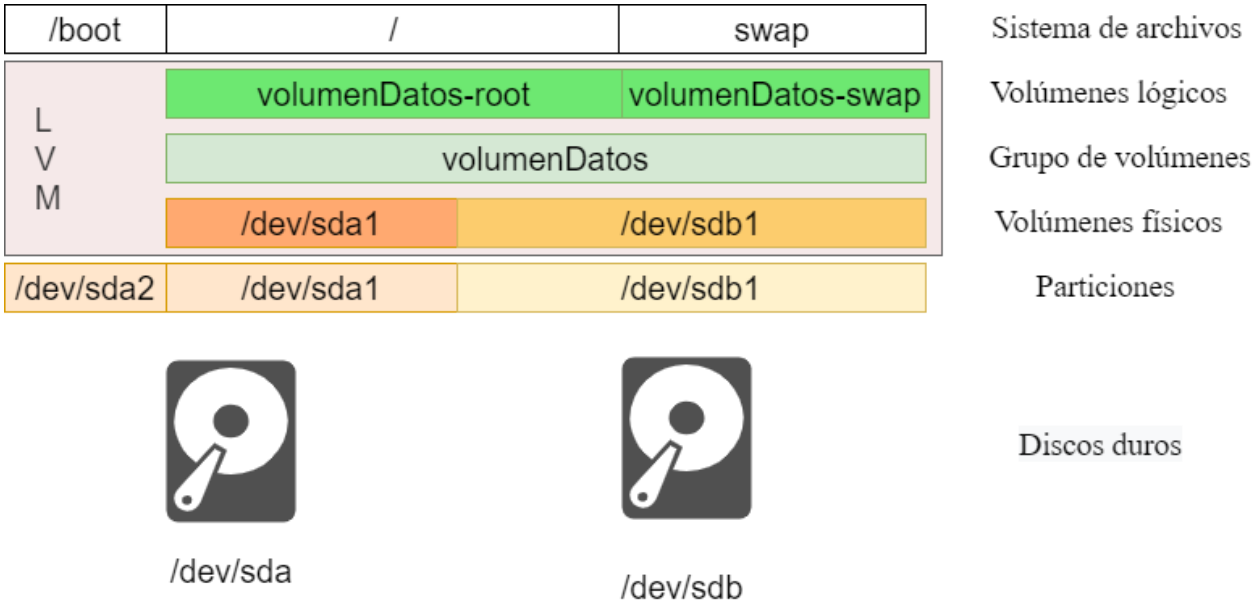

Ilustración B-0-5 Escenario propuesto para clonación LVM

Script para el proceso de clonación

El siguiente script permite la clonación completa del escenario.

En el script se realizan los siguientes pasos:

9. Clonar los datos de las particiones con datos.
10. Copiar el MBR de los discos.
11. Copiar tabla de particiones de los discos.
12. Hacer la copia de la configuración de LVM.
13. Guardar los UUID de los volúmenes físicos.
14. Guardar UUID y la etiqueta de la partición swap.

```
#!/bin/sh
set -e
echo "Comienza el proceso de clonacion"
echo "Realizando la copia de las particiones con datos"
fsarchiver -j2 -v savefs root_backup /dev/mapper/volumenDatos-root
fsarchiver -j2 -v savefs boot_backup /dev/sda2

echo "Copiando el MBR "
dd if=/dev/sda of=backup_sda_MBR.dd bs=512 count=1
dd if=/dev/sdb of=backup_sdb_MBR.dd bs=512 count=1

echo "Copiando tabla de particiones"
sfdisk -d /dev/sda > backup_sda.sfdisk
sfdisk -d /dev/sdb > backup_sdb.sfdisk
```

```
echo "Hacemos backup configuracion de LVM"
vgcfgbackup -f volumenDatosConfig volumenDatos

echo " Guardamos uuid de /dev/sda1"
for i in $(blkid /dev/sda1); do
case "$i" in
UUID*)
echo "$i" |sed "s/UUID/UUID_SDA1/g" > uuid_sda1 ;;
esac
done

echo " Guardamos uuid de /dev/sdb1"
for i in $(blkid /dev/sdb1); do
case "$i" in
UUID*)
echo "$i" |sed "s/UUID/UUID_SDB1/g" > uuid_sdb1 ;;
esac
done

echo " Guardamos uuid y label de swap"
for i in $(blkid /dev/mapper/volumenDatos-swap); do
case "$i" in
UUID*)
echo "$i" |sed "s/UUID/UUID_SWAP/g" > uuid_swap ;;
LABEL*)
echo "$i" | sed "s/LABEL/LABE_SWAP/g" > label_swap ;;
esac
done

echo "Terminado el proceso de clonacion correctamente"
```

Script para el proceso de restauración

El siguiente script permite la restauración completa del escenario.

En el script se realizan los siguientes pasos:

10. Restaurar el MBR de los discos.
11. Restaurar tabla de particiones de los discos.
12. Restaurar los volúmenes físicos.
13. Restaurar el grupo de volúmenes.
14. Activar los volúmenes lógicos.
15. Restaurar la partición swap.
16. Restaurar las particiones con datos.
17. Reinstalar el GRUB.

```
#!/bin/sh
set -e
echo "Comienza el proceso de restauracion"
echo "Restaurando el MBR y tabla de particiones"
echo "Restaurando MBR "
dd if=backup_sda_MBR.dd of=/dev/sda bs=512 count=1
dd if=backup_sdb_MBR.dd of=/dev/sdb bs=512 count=1

echo "Restaurando tabla de particiones extendida"
sfdisk /dev/sda < backup_sda.sfdisk
sfdisk /dev/sdb < backup_sdb.sfdisk

echo "Restaurando el volumen fisico /dev/sda1"
. uuid_sda1
pvcreate -u "$UUID_SDA1" --restorefile volumenDatosConfig /dev/sda1

echo "Restaurando el volumen fisico /dev/sda1"
. uuid_sdb1
pvcreate -u "$UUID_SDB1" --restorefile volumenDatosConfig /dev/sdb1

echo "Restaurando el grupo de volumenes volumenDatos"
vgcfgrestore -f volumenDatosConfig volumenDatos

echo "Activamos los volumenes logicos"
vgchange -a y

echo "Restaurando partición swap"
. uuid_swap
. label_swap
mkswap      -U      "$UUID_SWAP"      -L      "${LABEL_SWAP:-"swap"}"
/dev/mapper/volumenDatos-swap

echo "Restaurando las particiones con datos"
fsarchiver        -j2        -v        restfs        root_backup.fsa
id=0,dest=/dev/mapper/volumenDatos-root
fsarchiver -j2 -v restfs boot_backup.fsa id=0,dest=/dev/sda2

echo "Reinstando el GRUB"
mkdir datos
mount /dev/mapper/centos-root datos/
mount /dev/sda1 datos/boot/
mount --bind /dev/ datos/dev
mount --bind /proc/ datos/proc
chroot datos /bin/bash << "EOT"
/sbin/vgchange -y a
/sbin/grub2-install /dev/sda
/sbin/grub2-install /dev/sdb
EOT

echo "Terminado el proceso de clonacion correctamente"
```

ANEXO C: COPIAS DE SEGURIDAD INCREMENTALES BASADO EN GIT

En este anexo se detallarán conceptos básicos necesarios para entender el funcionamiento y la forma de trabajar de git. Se detallarán una serie de scripts que permitirán automatizar el funcionamiento de las copias de seguridad incrementales con el uso de la herramienta cron. También se explicará cómo descargar e instalar la herramienta git-store-meta.pl, la cual permite conservar los metadatos de los archivos en el repositorio.

Conceptos básicos sobre el uso de git

En esta sección se detallará unos conceptos básicos sobre el funcionamiento y uso de la herramienta git.

Cuando se trabaja con git disponemos de 3 zonas diferenciadas:

- Directorio de Trabajo. Esta zona es donde se realizan cambios, se modifican archivos, se añade nuevos o se eliminan. No es más que el directorio del sistema que está bajo el control de la herramienta git.
- Área de preparación. En esta zona se pasan los cambios que se hayan seleccionado para que se confirmen en la siguiente confirmación de cambios, y de esta forma, pasen a formar parte del repositorio.
- Repositorio local. Almacena, una vez que se confirman los cambios, los archivos que forman el repositorio, y de los cuales se mantendrá el histórico, para poder recuperar versiones anteriores.

Además, si se trabaja con un repositorio remoto, se poseerá de una zona más.

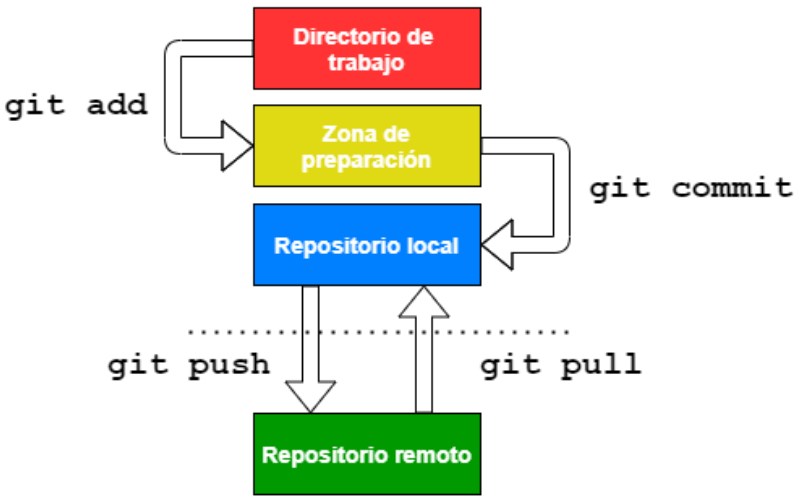

Ilustración C-0-1 Diagrama de zonas de un repositorio de git

El flujo más habitual cuando se usa la herramienta git es el siguiente:

1. Se realizan cambios en el directorio de trabajo.
2. Se añaden los cambios a la zona de preparación mediante el comando git add.
3. Se confirman los cambios que hubiera en la zona de preparación y se añaden al repositorio local con el comando git commit.

Si se tiene un repositorio remoto habría dos acciones más:

- La descarga de los cambios que hubiera en el repositorio remoto mediante el comando git pull.
- La subida de cambios que hubiera en el repositorio local al repositorio remoto mediante el comando git push.

Por último, en el flujo de trabajo de la herramienta git existe el concepto de ramas o bifurcaciones. Estos elementos son espacios donde el histórico de cambios se bifurca, permitiendo realizar modificaciones sin que afecten a otras ramas.

Una vez integrados todos los cambios en una rama, esta se puede fusionar con otra, de manera que la rama destino acoja todos los cambios. Esto es especialmente útil cuando varios usuarios pueden modificar el mismo repositorio. Normalmente se trabaja sobre la rama principal sobre la que se genera la versión. El resto de las ramas se utilizan para generar determinados cambios que requieran varias confirmaciones y que no se puedan integrar en la principal hasta que no estén completos. En este caso, solo se empleará la rama como un mecanismo para poder eliminar confirmaciones de cambios antiguas.

Instalación de la herramienta git-store-meta.pl

En esta sección se detallarán los requisitos y cómo instalar la herramienta git-store-meta.pl.

Las dependencias indicadas en el repositorio de Github son:

- Git version 1.7.2 o mayor.
- Plataforma Linux con entorno shell básico.
- Perl versión 5.8 o mayor con módulos integrados.
- Sort cualquier versión.
- Linux::moduleACL, o getfacl / setfacl (opcional, para manipular metadatos ACL).
- File::módulo lchown, o touch/chown (opcional, para aplicar metadatos a enlaces simbólicos).

El script para la instalación de la herramienta git-store-meta.pl será el siguiente

```
#!/bin/sh
cd /tmp/
git clone https://github.com/danny0838/git-store-meta
cd git-store-meta/
cp git-store-meta.pl /bin/
```

Script para la automatización de procedimientos

En esta sección se detallará como usar los scripts con los cuales se puede automatizar todos los procedimientos detallados en la sección Copias de seguridad incrementales basado en git.

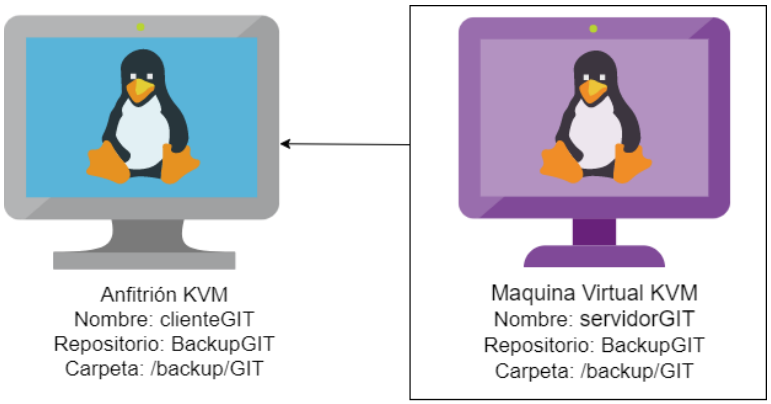

Ilustración C-0-2 Escenario propuesto para las copias de seguridad incrementales con git

Estos scripts proporcionarán junto a la memoria para su posterior uso.

Servidor

Los scripts que se detallarán a continuación permitirán configurar el servidor de git y programar las tareas de copiar los ficheros, realizar confirmaciones de cambios y llevar a cabo la eliminación de cambios antiguos y mantenimiento del repositorio.

Creación de repositorio

Se procederá a explicar el funcionamiento y los pasos que se van a realizar en el script configurarRepositorio.sh, además de cómo usarlo.

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobará las dependencias, las cuales son

- git.
- git-store-meta.pl.

A continuación, se le solicitará que introduzca los siguientes datos en orden:

1. Nombre Repositorio GIT
2. Ubicación de la carpeta para el repositorio de GIT
3. Introducir usuario propietario carpeta el repositorio de GIT.

Después de introducir todos los datos, se comenzará el procedimiento de creación y configuración del repositorio. A medida que se realicen las acciones se notificará al usuario. Si alguna fallara, se le notificará cual de ella ha sido, y, por lo tanto, dónde paró el procedimiento.

Una vez finalizado, se generan dos archivos configuracionRepo.cfg y repo.conf.

El fichero configuracionRepo.cfg contendrá una serie de variables necesarias para el resto de los scripts del servidor. Este indica cómo se ha configurado el repositorio. A continuación, se muestra un ejemplo:

```
USUARIO_REPOSITORIO=" usuarioBackup "
CARPETA_REPOSITORIO="/backup/GIT/"
NOMBRE_REPOSITORIO="BackupGIT"
```

El fichero repo.conf contendrá la información con la cual se ha configurado el repositorio. El contenido se presenta con un formato legible A continuación, se muestra un ejemplo:

```
Usuario propietario: usuarioBackup
Carpeta repositorio: /backup/GIT/
Nombre repositorio: BackupGIT
```

Preparación de ficheros para realizar la copia de seguridad

Se procederá a explicar el funcionamiento y los pasos que se realizan en el script copiarFicherosCarpetasServidor.sh, además de cómo usarlo.

Para hacer uso de este se debe tener en el mismo directorio el fichero configuracionRepo.cfg. Este se genera cuando se configura el repositorio. También se debe tener el fichero ficherosYCarptasRepositorio.cfg, que indicará cuáles son las carpetas o ficheros que se quieren incluir en el repositorio para realizar la copia de seguridad. Un ejemplo del fichero ficherosYCarptasRepositorio.cfg seria este:

```
# Ejemplo de fichero para realizar la copia en el repositorio
# Se debe hace uso de rutas absolutas
/var/lib/mysql/
/etc/
/var/lib/samba/dhcp.conf
```

Para comenzar, se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobará las dependencias, las cuales son:

- rsync.

A continuación, se procesará el fichero ficherosYCarptasRepositorio.cfg para eliminar los comentarios. Por cada línea se realizará la copia manteniendo los metadatos, ya sea del fichero o de los directorios.

Tras esto, se eliminarán los ficheros o directorios que estuvieran en el repositorio y que se han borrado del fichero ficherosYCarptasRepositorio.cfg porque no se quería mantener en las copias de seguridad.

Para finalizar, se eliminarán los archivos temporales creados a medida que se avanza en el procesamiento del script.

Se notificará al usuario a medida que se realizan las acciones. Si alguna fallara, se avisaría a este diciendo cual de todas ha provocado el error y, por lo tanto, donde se paró el procedimiento. Si existiera algún archivo temporal, se eliminaría antes de finalizar el script.

Todas las notificaciones de errores o del progreso del proceso se realizarán mediante el uso de la herramienta logger[121]. Todas ellas se pueden ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

Realización de la confirmación de cambios

Se procederá a explicar el funcionamiento y los pasos que se realizan en el script realizarCommitRepositorioServidor.sh y cómo usarlo.

Para hacer uso de este se debe tener en el mismo directorio el fichero configuracionRepo.cfg generado al configurar el repositorio.

Este script tiene como posibles argumentos:

-h	Mostrará un mensaje de ayuda del script. Es opcional.
-?	Mostrará un mensaje de ayuda del script. Es opcional.
-m "Mensaje"	Se proporciona un mensaje que se añadirá junto a la fecha como mensaje de confirmación de cambios. Un ejemplo de mensaje sería: Backup Mon Jun 00:00:00 1970 +0100 - Mensaje. Es opcional.

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobará las dependencias, las cuales son:

- git.
- git-store-meta.pl.

A continuación, se añadirán todos los cambios al área de preparación. Tras esto se comprobará si existen cambios que confirmar.

Si no existiera ningún cambio, se informará que no se relanzará la confirmación debido a que no existe cambios nuevos.

Si existiera algún cambio pendiente de confirmar, se realizará la confirmación de cambios cuyo mensaje de confirmación será la fecha de cuándo se llevó a cabo junto con mensaje que se indicó como argumento.

Se notificará al usuario a medida que se realizan las acciones. Si alguna fallara, se avisaría a este diciendo cual de todas ha provocado el error y, por lo tanto, donde se paró el procedimiento.

Todas las notificaciones de errores o del progreso del proceso se realizarán mediante el uso de la herramienta logger. Todas ellas se pueden ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

Reducción del histórico y optimización de repositorio

Se procederá a explicar el funcionamiento y los pasos que se llevan a cabo en el script borrarCommitAnterioresAFecha.sh, además de cómo usarlo.

Para hacer uso de este se debe tener en el mismo directorio el fichero configuracionRepo.cfg generado al configurar el repositorio.

Este script tiene argumento obligatorio:

'X Y ago'	Indica cuánto se debe retroceder para tomar la primera confirmación de cambios anterior a dicha fecha y borrar el histórico anterior a dicha confirmación de cambios. El formato es 'X Y ago' siendo Y month, year, day, hour, minute, second y X el número de divisiones de tiempo que se quiere retroceder.

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobará las dependencias, las cuales son:

- git.
- git-store-meta.pl.

A continuación, se buscará el primer de la primera confirmación de cambios posterior a la fecha calculada retrocediendo lo indicado en el argumento.

Tras esto se creará la rama temporal a partir de la confirmación de cambios y se hará la confirmación de cambio necesaria en la rama de la fecha calculada anteriormente y con el mensaje:

```
Truncado del historial antes de la fecha calculada
```

Se forzará el cambio del comienzo de la rama principal. Después, se borrará la rama temporal y se realizarán las tareas de mantenimiento del repositorio.

Se notificará al usuario a medida que se realizan las acciones. Si alguna fallara, se avisaría a este diciendo cual de todas ha provocado el error y, por lo tanto, donde se paró el procedimiento.

Todas las notificaciones de errores o del progreso del proceso se realizarán mediante el uso de la herramienta logger. Todas ellas se pueden ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

Cliente

Los scripts que se detallarán a continuación permitirán configurar el cliente de git, configurar el uso de la clave pública para acceder al servidor y programar tareas como actualizar el repositorio, copiar los ficheros en el repositorio y confirmar cambios.

Clonación del repositorio de git

Se procederá a explicar el funcionamiento y pasos que se realiza en el script configurarCliente.sh y como usarlo.

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobará las dependencias, las cuales son:

- git.
- git-store-meta.pl.
- ssh.

A continuación, se le solicitará que introduzca los siguientes datos en orden:

1. IP o Hostnames de servidor del repositorio a clonar.
2. Nombre Repositorio GIT.
3. Ubicación de la carpeta local para el repositorio de GIT.
4. Introducir usuario propietario carpeta local del repositorio de GIT.
5. Introducir usuario propietario repositorio de GIT.
6. Ubicación de la carpeta en el servidor para el repositorio de GIT.
7. Introducir puerto para ssh.
8. Desea usar clave pública para la configuración del repositorio(S/N).

Si se selecciona que se desea usar la clave pública, se solicitará que se seleccione una de las claves públicas que se encuentran en la carpeta /root/.ssh/.

Si se quiere usar una clave pública que no se ubique en la carpeta /root/.ssh/, se puede configurar manualmente o mediante el script configurarSshConexionConClavePublica.sh.

Después de introducir todos los datos, se comenzará el procedimiento de clonación de repositorio. Se notificará al usuario a medida que se realizan las acciones. Si alguna fallara, se avisaría a este diciendo cual de todas ha provocado el error y, por lo tanto, donde se paró el procedimiento.

Al finalizar, se generan dos archivos, configuracionRepoCliente.cfg y repo_cliente.conf.

El fichero configuracionRepoCliente.cfg contendrá una serie de variables necesarias para el resto de los scripts

del cliente y que indica cómo se ha configurado el repositorio. A continuación, se muestra un ejemplo:

```
  # Si se desea usar una clave primaria distinta de ~/.ssh/id_rsa hay
que declarar la variable  CLAVE_SSH
  #  CLAVE_SSH="~/.ssh/id_rsa"
  USUARIO_REPOSITORIO_LOCAL="usuarioBackup"
  CARPETA_REPOSITORIO_LOCAL="/backup/GIT/"
  NOMBRE_REPOSITORIO="BackupGIT"
  DIRECCION_REPOSITORIO="servidorGIT"
  CLAVE_SSH="/root/.ssh/id_rsa"
  PUERTO_SSH=22
```

El fichero repo_cliente.conf contendrá la información con la que se configura el repositorio adaptada en un formato legible. A continuación, se muestra un ejemplo:

```
  Usuario propietario: usuarioBackup
  Carpeta repositorio: /backup/GIT
  Nombre repositorio: BackupGIT
  Usuario remoto propietario: usuarioBackup
  Carpeta remota repositorio: /backup/GIT/
  Ubicacion remota: servidorGIT
  Puerto ssh usado para acceder mediante ssh con git: 22
  Clave publica usada para acceder mediante ssh con git:
/root/.ssh/id_rsa
```

Actualización del repositorio

Se procederá a explicar el funcionamiento y pasos que se realiza en el script actualizarCliente.sh.

Para hacer uso de este, se debe tener en el mismo directorio el fichero configuracionRepoCliente.cfg generado al configurar el repositorio.

Este script tiene como argumentos:

-h	Mostrará un mensaje de ayuda del script. Es opcional.
-?	Mostrará un mensaje de ayuda del script. Es opcional.
-m "Mensaje"	Se proporciona un mensaje que se añadirá junto a la fecha como mensaje de confirmación de cambios. Un ejemplo de mensaje sería: Backup Mon Jun 00:00:00 1970 +0100 - Mensaje. Es opcional.
-i	Indica que se ejecutara el script en modo iterativo si fuera necesario para realizar la autentificación de la conexión ssh. Si no se proporciona se utilizará lo configurado en configuracionRepoCliente.cfg.
-k clave_ssh	Indica el fichero que contiene la clave pública que se usará. Si no se proporciona se utilizará lo configurado en configuracionRepoCliente.cfg en la variable CLAVE_SSH.
-p puerto_ssh	Indica el puerto ssh que se usará. Si no se proporciona se utilizará lo configurado en

	configuracionRepoCliente.cfg en la variable PUERTO_SSH.

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Para hacer uso de este, se debe tener en el mismo directorio el fichero configuracionRepoCliente.cfg generado al configurar el repositorio.

Después el script comprobará las dependencias, las cuales son:

- git.
- git-store-meta.pl.
- ssh.

Primero se creará el ejecutable con la configuración para la conexión ssh. Se usa como en la variable de entorno GIT_SSH.

Luego se procederá a eliminar los posibles cambios que hubiera en el directorio de trabajo del repositorio del cliente para poder actualizar sin que haya conflicto con la información que hubiera en el servidor.

Para finalizar se actualizará la información del repositorio del cliente.

Se notificará al usuario a medida que se realizan las acciones. Si alguna fallara, se avisaría a este diciendo cual de todas ha provocado el error y, por lo tanto, donde se paró el procedimiento.

Todas las notificaciones de errores o del progreso del proceso se realizarán mediante el uso de la herramienta logger[121]. Todas ellas se pueden ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

Preparación de ficheros para realizar la copia de seguridad

Se procederá a explicar el funcionamiento y pasos que se realiza en el script copiarFicherosCarpetasCliente.sh.

La funcionalidad de este script es la misma que la de copiarFicherosCarpetasServidor.sh; la única diferencia es que usará el fichero configuracionRepoCliente.cfg en vez del fichero configuracionRepo.cfg.

Primero se creará el ejecutable con la configuración para la conexión ssh. Se usa como en la variable de entorno GIT_SSH.

A continuación, se eliminarán los posibles cambios que pudieran existir en el repositorio, y posteriormente, se descargarán los cambios que hubiera el servidor.

Se elimina el ejecutable con la configuración para la conexión ssh.

Se notificará al usuario a medida que se realizan las acciones. Si alguna fallara, se avisaría a este diciendo cual de todas ha provocado el error y, por lo tanto, donde se paró el procedimiento.

Todas las notificaciones de errores o del progreso del proceso se realizarán mediante el uso de la herramienta logger. Todas ellas se pueden ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

Realización de la confirmación de cambios

Se procederá a explicar el funcionamiento y pasos que se realiza en el script realizarCommitRepositorioServidor.sh y como usarlo.

Para hacer uso de este se debe tener en el mismo directorio el fichero configuracionRepoCliente.cfg generado al configurar el repositorio.

Este script tiene como posibles argumentos:

-h	Mostrará un mensaje de ayuda del script. Es opcional.
-?	Mostrará un mensaje de ayuda del script. Es opcional.
-m "Mensaje"	Se proporciona un mensaje que se añadirá junto a la fecha como mensaje de confirmación de cambios. Un ejemplo de mensaje sería: Backup Mon Jun 00:00:00 1970 +0100 - Mensaje. Es opcional.
-i	Indica que se ejecutara el script en modo iterativo si fuera necesario para realizar la autentificación de la conexión ssh. Si no se proporciona se utilizará lo configurado en configuracionRepoCliente.cfg.
-k clave_ssh	Indica el fichero que contiene la clave pública que se usará. Si no se proporciona se utilizará lo configurado en configuracionRepoCliente.cfg en la variable CLAVE_SSH.
-p puerto_ssh	Indica el puerto ssh que se usará. Si no se proporciona se utilizará lo configurado en configuracionRepoCliente.cfg en la variable PUERTO_SSH.

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobará las dependencias, las cuales son:

- git.
- git-store-meta.pl.
- ssh.

Primero se creará el ejecutable con la configuración para la conexión ssh que se usado como en la variable de entorno GIT_SSH.

A continuación, se añadirá todos los cambios al área de preparación y tras esto se comprobará si existiera cambios a confirmar.

Si no existirá ningún cambio se informará que no se relazará la confirmación debido a que no existe cambios

Si existirá algún cambio pendiente de confirmar, se realizará la confirmación de cambios cuyo mensaje de confirmación será la fecha de cuando se realizó junto al mensaje que se indicó como argumento.

Se elimina el ejecutable con la configuración para la conexión ssh.

Se notificará al usuario a medida que se realizan las acciones. Si alguna fallara, se avisaría a este diciendo cual de todas ha provocado el error y, por lo tanto, donde se paró el procedimiento.

Todas las notificaciones de errores o del progreso del proceso se realizarán mediante el uso de la herramienta logger. Todas ellas se pueden ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

Configuración de clave publica para la conexión ssh

Se procederá a explicar el funcionamiento y pasos que se realiza en el script configurarSshConexionConClavePublica.sh y como usarlo.

Este script permite configurar la conexión ssh mediante clave pública para acceder al sistema remoto donde se encuentra el servidor de git. Todo ello se lleva a cabo con el uso de los scripts de cliente, sin intervención humana. Si no se proporciona una clave pública, se generará una nueva. También se comprobará que no tuviera passpharse la clave pública proporcionada.

Para hacer uso de este, se debe tener en el mismo directorio el fichero configuracionRepoCliente.cfg generado al configurar.

Este script tiene como posibles argumentos:

-h	Mostrará un mensaje de ayuda del script. Es opcional.
-?	Mostrará un mensaje de ayuda del script. Es opcional.
-k key_propia	Indica la key que se usará para acceder a host_remoto con usuario_remoto. Si no se especifica, se buscará en ~/.ssh/, o se creará una con el nombre claveSsh sin passphrase.
- H host_remoto	Indica el host para el que se desea configurar el acceso mediante clave pública. Si no se proporciona se utilizará lo configurado en configuracionRepoCliente.cfg en la variable DIRECCION_REPOSITORIO.
-u usuario_remoto	Indica el usuario del host remoto. Si no se proporciona, se utilizará lo configurado en configuracionRepoCliente.cfg en la variable USUARIO_REPOSITORIO_SERVIDOR.
-p puerto_remoto	Indica el puerto ssh del host remoto. Si no se proporciona, se utilizará lo configurado en configuracionRepoCliente.cfg en la variable PUERTO_SSH.
-e	Indica que se modificará el fichero configuracionRepoCliente.cfg y repo_cliente.conf con la información usada en el script.

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobará las dependencias, las cuales son:

- ssh,
- ssh-keygen.
- ssh-copy-id.

Se comprobará si con los argumentos proporcionados es posible acceder mediante ssh, ya sea porque ya está configurado para la clave pública proporcionada o porque existe una configuración para el usuario en el sistema. Pueden darse tres situaciones:

- Si se ha proporcionado una clave pública pero no es posible acceder, se comprobará que no se tuviera passphrase. Tras esto, será necesario introducir la contraseña del usuario remoto.
- Si no fuera posible acceder y no se hubiera proporcionado ninguna clave pública, se buscará en el directorio /root/.ssh/ la existencia de alguna. Después, se solicitará seleccionar una de las encontradas. Tras esto, será necesario introducir la contraseña del usuario remoto.
- Si no fuera posible acceder y no se hubiera proporcionado ninguna clave pública y no existiera ninguna en el directorio /root/.ssh/, se creará una nueva con el nombre claveSsh. Tras esto, será necesario introducir la contraseña del usuario remoto.

Una vez configurado, si se indicó que se modificaría configuracionRepoCliente.cfg y repo_cliente.conf, se cambiará la información que indica el puerto ssh y la clave pública que usar.

Se notificará al usuario a medida que se realizan las acciones. Si alguna fallara, se avisaría a este diciendo cual de todas ha provocado el error y, por lo tanto, donde se paró el procedimiento. Si se hubieran creado archivos se eliminarán antes de finalizar el script.

Automatización de los procedimientos mediante el uso de cron

En esta sección se detallará un ejemplo de cómo realizar la configuración del demonio de cron para realizar la ejecución de los scripts de forma automática.

Para ello se hará uso de la herramienta crontab[121] que permite generar los ficheros necesarios para el demonio cron de manera sencilla.

En el cliente se ejecutará el comando `crontab -e` y se editará el fichero para incluir el siguiente fragmento:

```
  30 5 * * * /opt/clienteGIT/actualizarCliente.sh
```

Esto fragmento indica que se ejecute todos los días a las 5:30 el script actualizarCliente.sh.

En el servidor se ejecutará el comando `crontab -e` y se editará el fichero para incluir el siguiente fragmento:

```
  0 2 * * * /opt/servidorGIT/copiarFicherosCarpetasServidor.sh
  0 4 * * * /opt/servidorGIT/realizarCommitRepositorioServidor.sh
  0 0 31 12 * /opt/servidorGIT/borrarCommitAnterioresAFecha.sh '2
year ago'
```

Esto fragmento indica las siguientes acciones:

- Se ejecuta todos los días a las 2:00 el script copiarFicherosCarpetasServidor.sh.
- Se ejecuta todos los días a las 4:00 el script realizarCommitRepositorioServidor.sh.

Se ejecuta el 31 de diciembre a las 00:00 el script borrarCommitAnterioresAFecha.sh con el argumento '-2 year ago', el cual indica que se eliminarán las confirmaciones de cambios que tengan más de dos años.

ANEXO D: SINCRONIZACIÓN DE COPIAS DE SEGURIDAD MEDIANTE RSYNC

En este anexo se detallará como usar los scripts con los cuales se puede automatizar todos los procedimientos detallados en el aparato Sincronización de copias de seguridad mediante rsync y como configurar su uso con la herramienta cron.

Estos scripts proporcionarán junto a la memoria para su posterior uso.

Servidor

Se procederá a explicar el funcionamiento y los pasos que se llevan a cabo en el script scriptBackupCrearChecksumYConvertirInmutable.sh, además de cómo usarlo.

Para hacer uso de este, se debe tener en el mismo directorio el fichero backupConfig.cfg. Este fichero contiene la configuración para el uso del script.

```
  #backupConfig.cfg
  #Archivo de configuracion para el script
scriptBackupCrearChecksumYConvertirInmutable.sh

  # Usuario que propietario de la carpeta de CARPETA_BACKUP
  USUARIO_BACKUP="usuarioBackup"

  # Carpeta donde se encuentra las copias de seguridad
  ## Debe ser la ruta absoluta terminado /
  CARPETA_BACKUP="/backup/Rsync/"

  # Carpeta donde se almacenaran los ckecksum de las carpetas copiadas
  ## La ruta completa será $CARPETA_BACKUP$CARPETA_CHECKSUM
  ## Debe terminar con /
  CARPETA_CHECKSUM="ckecksum/"

  # Prefijo comun en las carpetas que contienen las diferencias
seguridad
  PREFIJO_BACKUP="copiaSeguridad_"
```

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobará las dependencias, las cuales son:

7. tar
8. md5sum
9. chattr

Luego, se consultarán todos los directorios que comiencen por el prefijo $PREFIJO_BACKUP. El proceso será cambiar el usuario propietario de los ficheros y directorios para después convertirlos en inmutables.

Tras esto, se calculará la suma de comprobación de cada directorio que comience por el prefijo $PREFIJO_BACKUP. Será almacenada en un fichero con el nombre del directorio en la carpeta

$CARPETA_CHECKSUM. Si existiera ya una suma de comprobación calculada para el directorio, se validará si esta ha cambiado, y se notificará al usuario de la modificación de los archivos y directorios. Así, finalizaría el script.

Para finalizar, se convertirán en inmutables y se les cambiará el usuario propietario a las sumas de comprobación calculadas.

Se notificará al usuario a medida que se realizan las acciones. Si alguna fallara, se avisaría a este diciendo cual de todas ha provocado el error y, por lo tanto, donde se paró el procedimiento.

Todas las notificaciones de errores o del progreso del proceso se realizarán mediante el uso de la herramienta logger. Todas ellas se pueden ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

A continuación, se muestra un ejemplo de configuración con la herramienta cron para el caso de que se ejecute el script de forma periódica. Se ejecutará el comando `crontab -e` y se editará el fichero para incluir el siguiente fragmento:

```
  0 7 * * SAT
/opt/servidorRsync/scriptBackupCrearChecksumYConvertirInmutable.sh
```

Este fragmento indica que se lanzará el script los sábados a las 7:00.

Cliente

Se procederá a explicar el funcionamiento y los pasos que se llevan a cabo en el script scriptBackupRemoto.sh, además de cómo usarlo.

Para hacer uso de este, se debe tener en el mismo directorio el fichero backupConfigRemoto.cfg. Este fichero contiene la configuración para el uso del script. El contenido del fichero será:

```
  #backupConfigRemoto.cfg
  #Archivo de configuracion para el script scriptBackupRemoto.sh

  # Usuario que propietario de la carpeta de $CARPETA_BACKUP
  USUARIO_BACKUP="usuarioBackup"

  # Usuario que propietario de la carpeta de $CARPETA_BACKUP_SERVIDOR
en el servidor
  USUARIO_BACKUP_SERVIDOR="usuarioBackup"

  # Carpeta donde se almacenra las copias de seguridad
  ## Debe ser la ruta absoluta terminado /
  CARPETA_BACKUP="/backup/Rsync/"

  # Carpeta donde se encuentra las copias de seguridad
  ## Debe ser la ruta absoluta terminado /
  CARPETA_BACKUP_SERVIDOR="/backup/Rsync/"

  # Carpeta donde se almacenaran los ckecksum de las carpetas copiadas
  CARPETA_CHECKSUM="ckecksum/"

  # Carpeta donde se almacenaran los ckecksum de las carpetas copiadas
en el servidor
  CARPETA_CHECKSUM_SERVIDOR="ckecksum/"
```

```
  # Prefijo comun en las carpetas que contienen las diferencias
seguridad
  PREFIJO_BACKUP="copiaSeguridad_"

  # Sera la dirección remota donde se recuperan las copias
  DIRECCION_REMOTA_BACKUP="servidor01"

  # Ubicacion del binario de ssh
  SSH_BIN="/bin/ssh"

  # ssh puerto
  SSH_PORT="22"

  # Ruta abdoluta de la clave publica usada para la conexion ssh
  ## Opcional de no especificarse la conexion ssh se hara sin el uso
de clave publica
  ## La clave publica debe no tener passpharse
  PUBLIC_KEY="~/.ssh/id_rsa"
```

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobará las dependencias, las cuales son:

10. ssh
11. rsync
12. tar
13. md5sum
14. chattr

Se consultarán todos los directorios que comience por el prefijo $PREFIJO_BACKUP en el directorio $CARPETA_BACKUP_SERVIDOR del servidor indicado por $DIRECCION_REMOTA_BACKUP.

Tras esto se comprobará si los directorios fueron modificaron en el servidor a través de recalcular la suma de comprobación y de compararla con la almacenada en $CARPETA_CHECKSUM_SERVIDOR. Si algún directorio fue cambiado, se buscará la existencia de una copia en el cliente. También se recalculará la suma de comprobación para dicho directorio con la información del cliente y se comparará con la almacenada en $CARPETA_CHECKSUM. Se notificará al usuario de la existencia de una copia de seguridad modificada que podrá ser sustituida por la copia existente en el cliente.

Después de comprobar la integridad de las copias de seguridad en el servidor y de que no haya ninguna modificada, se realizará la transferencia de las copias que no se encontraban en el cliente. Una vez terminada la transferencia, se calcularán las sumas de comprobación de los directorios y se almacenarán en $CARPETA_CHECKSUM. Para finalizar, se convertirán en inmutables y se les cambiará el usuario propietario a las sumas de comprobación calculadas.

Se notificará al usuario a medida que se realizan las acciones. Si alguna fallara, se avisaría a este diciendo cual de todas ha provocado el error y, por lo tanto, donde se paró el procedimiento. Todas las notificaciones de errores o del progreso del proceso se realizarán mediante el uso de la herramienta logger. Todas ellas se pueden ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

A continuación, se muestra un ejemplo de configuración con la herramienta cron para el caso de que se ejecute el script de forma periódica. Se ejecutará el comando `crontab -e` y se editará el fichero para incluir el siguiente fragmento:

```
  0 7 * * SUN /opt/clienteRsync/scriptBackupRemoto.sh
```

Esto fragmento indica que se ejecute el script los domingos a las 7:00.

ANEXO E: CONFIGURACIÓN ILO

En este anexo se detallará las ventajas interesantes que ofrece la adquisición de una licencia y como configurar el uso de esta.

Activación de la licencia iLO Advanced

Los procesadores iLO que se encuentra en los servidores HP ProLiant poseen ciertas características que no son accesibles o se encuentran limitadas hasta que se active su firmware mediante una licencia de pago.

La adquisición de la licencia permite, entre otras características detalladas en la tabla 8 del manual de usuario de iLO:

- El uso de la consola remota integrada completa. Si no disponemos de la licencia no podremos usarlo una vez inicie el sistema operativo.
- Uso de dispositivos remoto, permite la conexión remota de unidades de almacenamiento o carpeta que serán accesible en el servidor como si se encontrará físicamente conectado. Solo accesible si se dispone licencia activa.
- Alerta mediante correo que solo es configurable si se dispone licencia activa.

Para realizar la configuración de la licencia es necesario seguir lo siguiente pasos:

1. Acceder a la página de web de iLO y una vez autentificado pulsar en la entrada del menú lateral Licensing en el submenú de Administration.

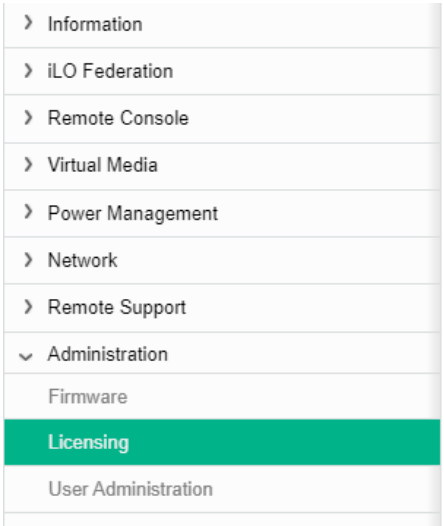

Ilustración E-0-1 Ubicación de la página de la licencia en el menú lateral de la web de iLO

2. Introducir la licencia adquirida en el campo Activation key.

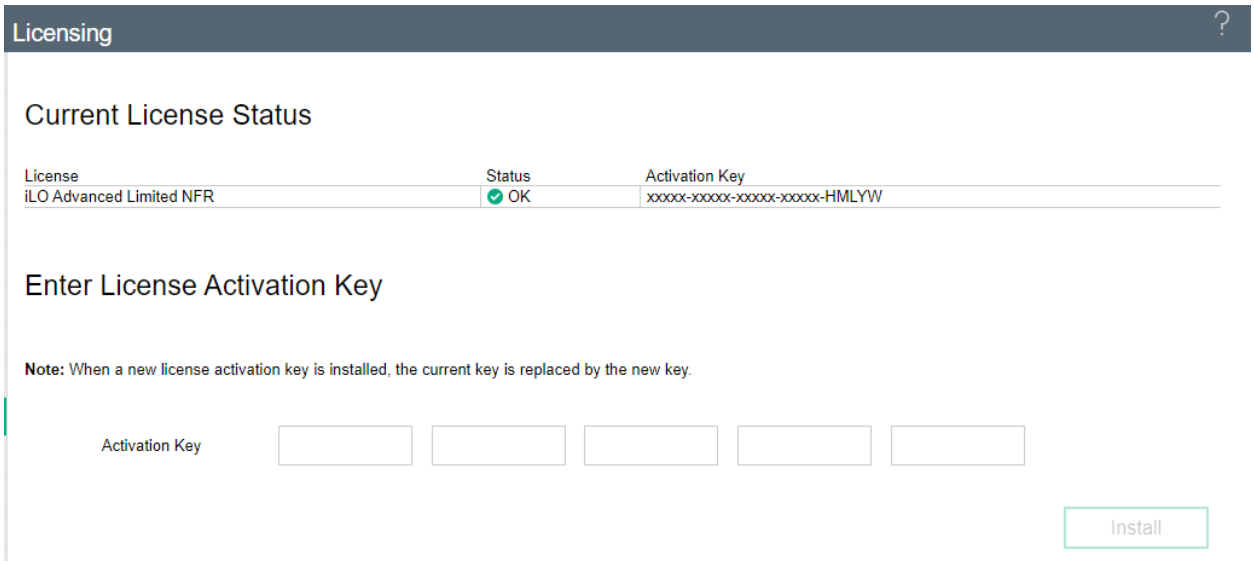

Ilustración E-0-2 Página de activación de licencia iLO

3. Pulsar el botón Install y aceptar el popup de advertencia.

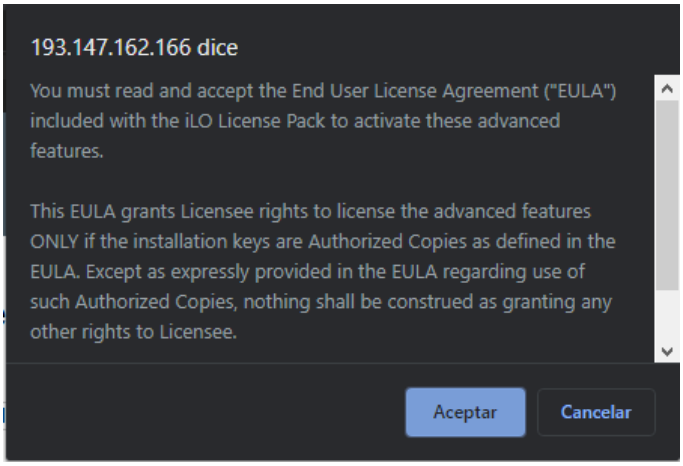

Ilustración E-0-3 Popup de confirmación activación de licencia iLO

ANEXO F: GESTIÓN DE MÁQUINAS VIRTUALES QEMU-KVM

En el anexo se detallará el procedimiento usado para realizar la exportación o importación de máquinas virtuales, el uso de los diferentes scripts que componen los menús y la estructura del fichero de configuración de los scripts.

Proceso de exportación de máquinas virtuales QEMU-KVM.

Para la realización de este proceso se proporcionará los siguientes tres scripts junto a la memoria:

- exportarMV.sh, que permite realizar el proceso desde línea de comando y como argumento es necesario indicar el nombre de la máquina a exportar.
- exportarMVdesatentido.sh, que realizar el proceso de forma totalmente desatendida.

Este script tiene como posibles argumentos:

-h	Mostrará un mensaje de ayuda del script. Es opcional.
-?	Mostrará un mensaje de ayuda del script. Es opcional.
-o nombreOriginal	Indica el nombre de la maquina a exportar.
-n nombreNuevo	Indica el nuevo nombre de la maquina tras exportarla.
-d rutaDestino	Indica donde se realizar la exportación y debe ser una ruta absoluta.
-c	Indica que se desea comprimir tras exportar por defecto no se comprimirá.
-x dicosAExcluir	Indica los discos que no se quieren incluir al exportar. El formato del argumento tiene que ser el parámetro que aparece en el XML en las etiquetas de los discos <source "/"file='*'/> para cada disco separado por espacio y entrecomillado.

De no introducir alguno de los argumentos, excepto los que muestran el mensaje de ayuda, se recurrirá al fichero de configuración configuracionScriptVirsh.cfg.

- exportarMVMenu.sh, que realizar el proceso de a través de menús generado con la herramienta dialog y como argumento es necesario indicar el nombre de la máquina a exportar.

Los pasos que se realizan en el proceso son:

1. Comprobar las dependencias con las herramientas virsh, tar[122] y si se hacen uso de menús también de la herramienta dialog.
2. Comprobar que se ha iniciado la sesión con el usuario root.
3. Comprobar que la máquina virtual de origen existe.
4. Comprobar que la carpeta destino donde se almacenará la maquina exportada.
5. Comprobar si la máquina virtual se encuentra activa y de estarlo la apagará.
6. Extraer el fichero XML que define la máquina virtual a la carpeta destino.
7. Listar los discos asociado a la máquina virtual.

8. Permitir al usuario seleccionar aquellos discos que se desea exportar.
9. Copiar los discos seleccionado a la máquina virtual a la carpeta destino.
10. Generar el fichero discos_Asociado_nombreMaquinaVirtual donde se almacenará los discos que han sido exportado y cuales nos. El formato del fichero es el siguiente:

```
/var/kvm-VMs/ait08/ait08-HD-Raiz.qcow2|COPY
/var/kvm-VMs/ait08/ait08-HD-Web.qcow2|COPY
/var/kvm-VMs/ait08/ait08-HD-Exportado_NFS.qcow2|COPY
/var/kvm-VMs/ait08/ait08-HD-Backup.qcow2|COPY
/var/kvm-VMs/ait08/ait08-HD-Almacen.qcow2|COPY
/var/kvm-VMs/ait08/ait08-HD-OpenGnSys_Imagenes.qcow2|NO_COPY
/var/kvm-VMs/ait08/ait08-HD-Swap.qcow2|NO_COPY
```

Donde los discos que no han sido exportado serían los dos últimos.

11. Eliminar el UUID de la máquina virtual y las direcciones macs asociadas a la máquina virtual del fichero XML extraído.

    Esto permite que cuando se regenere la máquina virtual a través del fichero XML tanto el UUID como las direcciones macs se calculen de nuevo para que sean únicos en el sistema. Para realizar estos eliminamos todos los elementos UUID o mac address que exista en el fichero XML
12. Renombrar la maquina exportada para asignarle otro nombre si el usuario lo indico.
13. Comprimir los ficheros en a la carpeta destino que componen la exportación de la maquina si el usuario lo indico.
14. Reanudar la máquina virtual si se encontraba activa antes del proceso.

Proceso de importación de máquinas virtuales QEMU-KVM.

Para la realización de este proceso se proporcionará los siguientes tres scripts junto a la memoria:

- importarMV.sh, que permite realizar el proceso desde línea de comando y como argumento es necesario indicar el nombre de la máquina a exportar.
- importarMVdesatentido.sh, que realizar el proceso de forma totalmente desatendida.

Este script tiene como posibles argumentos:

-h	Mostrará un mensaje de ayuda del script. Es opcional.
-?	Mostrará un mensaje de ayuda del script. Es opcional.
-o rutaOrigen	Indica donde se encontrará los archivos necesarios para importar la máquina.
-n nombreNuevo	Indica el nuevo nombre de la maquina a importar.
-d rutaDestino	Indica donde se copiará los discos tras la importación.
-f fichero	Indica el fichero comprimido se usará como origen para la importación (deberá ser la ruta absoluta).
-x	Indica si se usara el un archivo comprimido para importar por defecto no se usará

De no introducir alguno de los argumentos, excepto los que muestran el mensaje de ayuda, se recurrirá al fichero de configuración configuracionScriptVirsh.cfg.

- importarMVMenu.sh, que realizar el proceso de a través de menús generado con la herramienta dialog y como argumento es necesario indicar el nombre de la máquina a exportar.

Los pasos que se realizan en el proceso son:

1. Comprobar las dependencias con las herramientas virsh, tar y si se hacen uso de menús también de la herramienta dialog.
2. Comprobar que se ha iniciado la sesión con el usuario root.
3. Comprobar que la máquina virtual de origen existe.
4. Comprobar que la carpeta destino donde se copiará los discos tras la importación.
5. Comprobar si el nombre de la nueva máquina virtual ya existe en el sistema.
6. Descomprimir si partimos de un fichero comprimido.
7. Comprobar que existe el XML que define la maquina a importar.
8. Copiar los discos asociados a la máquina virtual indicado en el fichero discos_Asociado_nombreMaquinaVirtual con COPY.
9. Modificar el XML para incluir la nueva ruta de los discos duros copiados.
10. Crear la nueva la nueva máquina virtual a partir del fichero XML.
11. Indicar los discos que no exportaron, se encuentra asociado a la máquina virtual con su antigua ruta y que es necesario modificar manualmente. Estos son los discos indicado en el fichero discos_Asociado_nombreMaquinaVirtual con NO_COPY.

Fichero de configuración configuracionScriptVirsh.cfg

El fichero configuracionScriptVirsh.cfg es usado en los scripts exportarMV.sh, exportarMVdesatentido.sh, exportarMVMenu.sh, importarMV.sh, importarMVdesatentido.sh y importarMVMenu.sh, menuVirshCompleto.sh. En cada uno de ello es usado un fragmento de este.

A continuación, se detallarán diferentes fragmentos del fichero de configuración:

- Fragmento común entre los scripts.

```
#Archivos usado para almacenar información temporal de la ejecución
de los scripts
OUTPUT="/tmp/output"
ERROR_COMPRIMIR="/tmp/error"
ERROR_DESCOMPRIMIR="/tmp/error"
```

- Fragmento usado en el script menuVirshCompleto.sh.

```
#Indica que editor se usará para editar el XML de las maquinas
EDITOR_POR_DEFECTO=nano
```

- Fragmento usado en el script exportarMVMen.sh.

```
#Esta indica la carpeta que saldran en la seleccion por defecto para
almacenar los archivos de la exportacion de la maquina
  DIRECTORIO_POR_DEFECTO_DESTINO=$PWD"/KVMcopia/"
```

- Fragmento usado en el script importarMVMenu.sh.

```
 #Esta indica la carpeta que saldrán en la selección por defecto
como origen de los ficheros comprimido de la máquina a importar
DIRECTORIO_POR_DEFECTO_ORIGEN=$PWD"/KVMcopia/"

 #Esta indica la carpeta que saldrán en la selección por defecto
para almacenar los discos de la maquina a importar
DIRECTORIO_POR_DEFECTO_DESTINO_DISCO=$PWD"/KVMcopia/"
```

- Fragmento común importarMVdesatentido.sh y expotarMVdesatentido.sh.

```
#Indica el nombre de la máquina origen y destino que se tomaran para
exportar e importar en modo desatendido
VM_ORIGINAL="nombremaquinaAntiguo"
VM_NUEVA="nombremaquinaNuevo"
```

- Fragmento usado en el script expotarMVdesatentido.sh.

```
#Indica si se quieren partir generar un archivo comprimido con los
ficheros de una máquina
COMPRIMIR="FALSE"

#Indica si se va a desear seleccionar los discos a exportar, si está
a FALSE se exportarán todos
SELECCIONAR_DISCOS="TRUE"

#Indica los discos que no se exportaran al hacer uso de la variable
SELECCIONAR_DISCOS a TRUE y del script modo desatendido
DISCOS_A_EXCLUIR="" # Introducir el parámetro que aparece en el XML
la etiqueta del disco < source "/"file='*'/> separado por espacio

#Esta indica la carpeta donde se almacenará los archivos de la
exportacion de la maquina en modo desatendido
CARPETA_SELECCIONADA_DESANTENDIDO="/var/backup/"
```

- Fragmento usado en el script importarMVdesatentido.sh.

```
#Indica si se quieren partir de un archivo comprimido en tar.gz para
importar una máquina
DESCOMPRIMIR="FALSE"

#Esta indica la carpeta que contiene los archivos no comprimido de
la máquina a importar que se usara en modo desatendido
DIRECTORIO_ORIGEN_MAQUINA_DESANTENDIDO="/home/dit/"

#Esta indica la carpeta destino de los discos de la máquina a
importar que se usara en modo desatendido
CARPETA_UBICACION_DISCOS_DESANTENDIDO="/home/dit/KVMcopia/"$VM_NUE
VA"/"

#Esta indica el fichero que se usara en modo desatendido como origen
comprimido de la máquina a importar
ARCHIVO_COMPRIMIDO_DESANTENDIDO="fichero.tar.gz"
```

ANEXO G: CLÚSTER DE ALTA DISPONIBILIDAD

En el anexo se detallará la configuración de los dispositivos de bloques creados con drbd y el funcionamiento de script scriptReglaIptables.sh que permite la creación o eliminación de reglas iptables mediante el uso del agente de recurso anything en el clúster con el objetivo de redirigir las peticiones al nodo donde se encuentre un recurso.

Configuración de los dispositivos de bloques mediante drbd.

El archivo de configuración /etc/drbd.d/recursos.res este compuesto por tres dispositivos distintos y lo cuales su propiedad son la mismas.

```
resource imagenes {
    meta-disk internal;
    device /dev/drbd1;
    disk /dev/sda3;
    syncer {
     rate 80M;
    }
    net {
     allow-two-primaries;
     after-sb-0pri discard-zero-changes;
     after-sb-1pri discard-secondary;
     after-sb-2pri disconnect;
    }
    on servidor1 { address IP1:7789; }
    on servidor2 { address IP2:7789; }
}

resource basedatos {
    meta-disk internal;
    device /dev/drbd2;
    disk /dev/sda6;
    syncer {
     rate 80M;
    }
    net {
     allow-two-primaries;
     after-sb-0pri discard-zero-changes;
     after-sb-1pri discard-secondary;
     after-sb-2pri disconnect;
   }
    on servidor1 { address IP1:7790; }
    on servidor2 { address IP2:7790; }
}

resource tftp {
    meta-disk internal;
    device /dev/drbd3;
```

```
    disk /dev/sda7;
    syncer {
     rate 80M;
    }
    net {
     allow-two-primaries;
     after-sb-0pri discard-zero-changes;
     after-sb-1pri discard-secondary;
     after-sb-2pri disconnect;
    }
    on servidor1 { address IP1:7791; }
    on servidor2 { address IP2:7791; }
}
```

A continuación, se detallará las diferentes propiedades:

resource X	Establece el nombre del recuso y por el que se identificará para ser administrado.
meta-disk internal	Indica que los metadatos se almacenarán al final del dispositivo.
device /dev/drbdX	Indica el nombre del dispositivo que se creará con este recurso.
disk /dev/sdaX	Indica la partición que se usará para crear el dispositivo de bloque
rate 80M	Establece el límite de consumo de ancho de banda en 80Mb/s.
allow-two-primaries	Permite que dos nodos actúen como nodos primarios para el recurso.
after-sb-0pri discard-zero-changes	Indica que al detectar una desincronización entre los nodos y ninguno de ellos se encuentra como nodo primario, se descartara los contenidos de un nodo si no se detectaron cambios desde que la desincronización.
after-sb-1pri discard-secondary	Indica que al detectar una desincronización entre los nodos y uno de ellos se encuentra como nodo primario, se descartará los contenidos del nodo secundario.
after-sb-2pri disconnect	Indica que al detectar una desincronización entre los nodos y ambos de ellos se encuentra como nodo primario, se desconectará la sincronización y deberá solucionarse manualmente.
on servidorX { address IPX:Puerto; }	Declara en que nodo se creará el dispositivo de bloque y para ello se declara un nombre para el nodo, su dirección IP y el puerto que se usará para la comunicación entre los nodos. El puerto debe cambiar entre diferentes recursos

Script para la generación de la regla de redirección de peticiones a un recurso del clúster.

El script scriptReglaIptables.sh crea reglas de iptables que hará se redirige el tráfico de los puertos especificado hacia el nodo donde se encuentra alojado el recurso del clúster actualmente para poder hacer al recurso a través de la IP de cualquier nodo al recurso que se desea.

Está pensado para hacer uso del recurso ocf:heartbeat:anything en un clúster para llamar a este script y debe ejecutarse en los nodos que no que estén ejecutando el recurso.

Este script tiene como posibles argumentos:

-h	Mostrar ayuda. Es opcional.
-?	Mostrar ayuda. Es opcional.
-d	Indica que se borran las reglas creada de iptables con los parámetros idRecuso y puertos y se finalizará. Es opcional.
-w segundosEspera	Indica los segundos de espera para comprobamos que el recurso siga activo en el mismo nodo desde que se inició el script. Es opcional.
-r idRecurso	Indica el recurso que se buscara para redireccionar los puertos parado como parámetros
-p puertos	Indica los puertos a redireccionar hacia el nodo donde se encuentra el recurso id Recurso. Algunos ejemplos son 20/udp 80/tcp 1900:1910/tcp 20/udp,21/tcp

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobará las dependencias, las cuales son:

- pcs
- xmllint[123]

Tras esto se comprobará que ha sido introducido el id de un recurso configurado en el clúster y que puerto se han de redireccionar, de no introducirse notificará un error y se terminará la ejecución con un error.

Luego se verificará que los protocolos introducido para los puertos sean únicamente tcp o udp, si se detecta que algún puerto no tuviera uno de esto protocolo asociado se notificará un error y se terminará la ejecución con un error.

El formato para el argumento que especifica el puerto puede ser uno de los siguientes:

- 20/udp: Indica que se quiere redirigir únicamente el puerto 20.
- 1900:1910/tcp: Indica que se quiere redirigir los puertos desde el 1900 al 1910 ambos inclusive.

Se permite concatenar varios de los formatos anteriores siempre y cuando sea se parado por coma.

A continuación, se comprueba que el recurso indicado existe se está ejecutando en algún nodo y de no estarlo se notificará un error y se detendrá la ejecución.

Se calcula la dirección IP del nodo donde se encuentra el recurso a partir del fichero de hosts, si no existiera se notificará un error y se terminará la ejecución con un error.

Se crean las reglas de iptables para la redirección, para ello se añade las siguientes reglas en la tabla de nat:

- Una regla en la cadena de PREROTING que indica que se haga DNAT contra la dirección IP donde se encuentra el recurso.

- Una regla en la cadena de POSTROUTING que indica que se realice MASQUERADE antes de enrutar el paquete.
- Una regla en la cadena de OUTPUT que indica que se haga DNAT contra la dirección IP donde se encuentra el recurso.

Antes de crear cada una de las reglas para cada puerto indica se comprueba que no exista una ya aplicada para que no se dupliquen las reglas.

Por último, realiza una espera comprobando cada un cierto tiempo que el recurso no ha cambiado de nodo o a dejado de ejecutarse en el clúster.

Si el recurso cambiara de nodo, dejará de ejecutarse o el script fuera detenido por el clúster o manualmente antes de terminar la ejecución del script se realizará un borrado de todas las reglas generada por el script y de esta manera eliminar la redirección de puertos realizada.

Todas las notificaciones de errores o del progreso del proceso se realizará mediante el uso de la herramienta logger y se puede ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

ANEXO H: ARRANQUE DE MÁQUINA VIRTUAL DE MANERA AUTOMÁTICA

En este anexo se detallará como poder configurar las máquinas virtuales para poder acceder ya sea mediante el protocolo RDP o VNC y se detallará la configuración usada para el gestor de ventana i3.

Configuración VNC en máquinas virtuales basadas en VMware

Para realizar la configuración del servidor VNC se puede realizar mediante la modificación de del fichero vmx que define a la máquina virtual o cambian la configuración mediante la interfaz gráfica.

Para activar el servidor VNC que modificará el fichero de configuración será necesario añadir las siguientes líneas.

```
RemoteDisplay.vnc.enabled = "true"
RemoteDisplay.vnc.port = "5999"
RemoteDisplay.vnc.password = "Password"
```

Con esta línea se estaría habilitando el servidor VNC, configurando el puerto de escucha en el 5999 y añadiendo la contraseña Password, esta última configuración es opcional.

Para realizar la activación vía interfaz gráfica se tendrá que acceder a la configuración de la máquina virtual y luego a la pestaña de opciones y al ajuste Conexiones VNC. Una vez allí se podrá realizar las mismas configuraciones que mediante la modificación del fichero vmx.

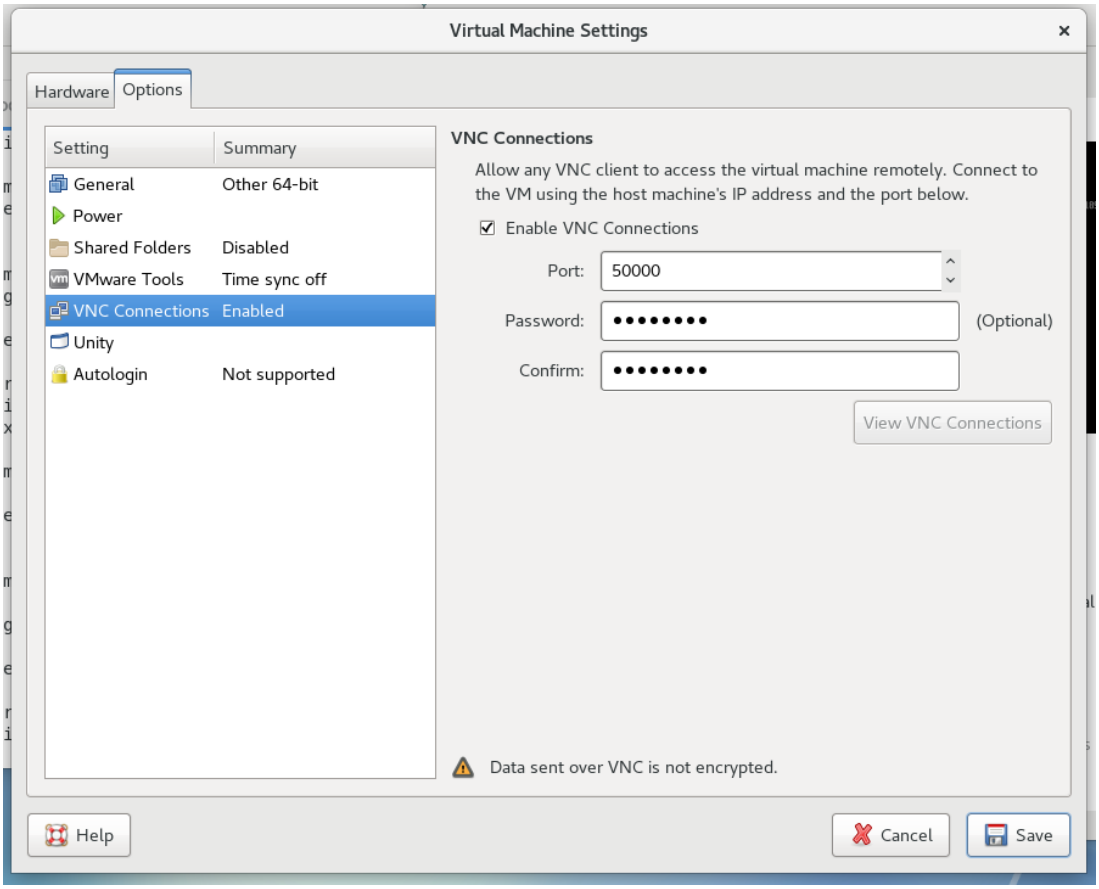

Ilustración H-0-1 Configuración VNC en VMware

Configuración RDP en máquinas virtuales basadas en VirtualBox

Para realizar la configuración del servidor RDP se puede realizar mediante la modificación de del fichero vbox que define a la máquina virtual o cambian la configuración mediante la interfaz gráfica.

Esta funcionalidad es parte del paquete VirtualBox Extension Pack y será necesario tenerlo instalado para poder utilizarla. Tambien es importante que en la configuración el método de autentificación sea nulo para poder acceder correctamente mediante el cliente freerdp.

Para activar el servidor RDP que modificará el fichero de configuración será necesario añadir las siguientes líneas dentro de la etiqueta Hardware.

```
  <RemoteDisplay           enabled="true"           authTimeout="2000"
allowMultiConnection="true">
          <VRDEProperties>
            <Property name="TCP/Ports" value="5999"/>
           </VRDEProperties>
  </RemoteDisplay>
```

Con esta línea se estaría habilitando el servidor RDP sin método de autentificación, configurando el puerto de escucha en el 5999, con tiempo de espera para autentificación 2000ms y permitiendo la múltiple conexión, estas dos últimas configuraciones son opcionales.

Para realizar la activación vía interfaz gráfica se tendrá que acceder a la configuración de la máquina virtual y luego al ajuste de pantalla y a la pestaña de Pantalla remota Conexiones VNC. Una vez allí se podrá realizar las mismas configuraciones que mediante la modificación del fichero vbox.

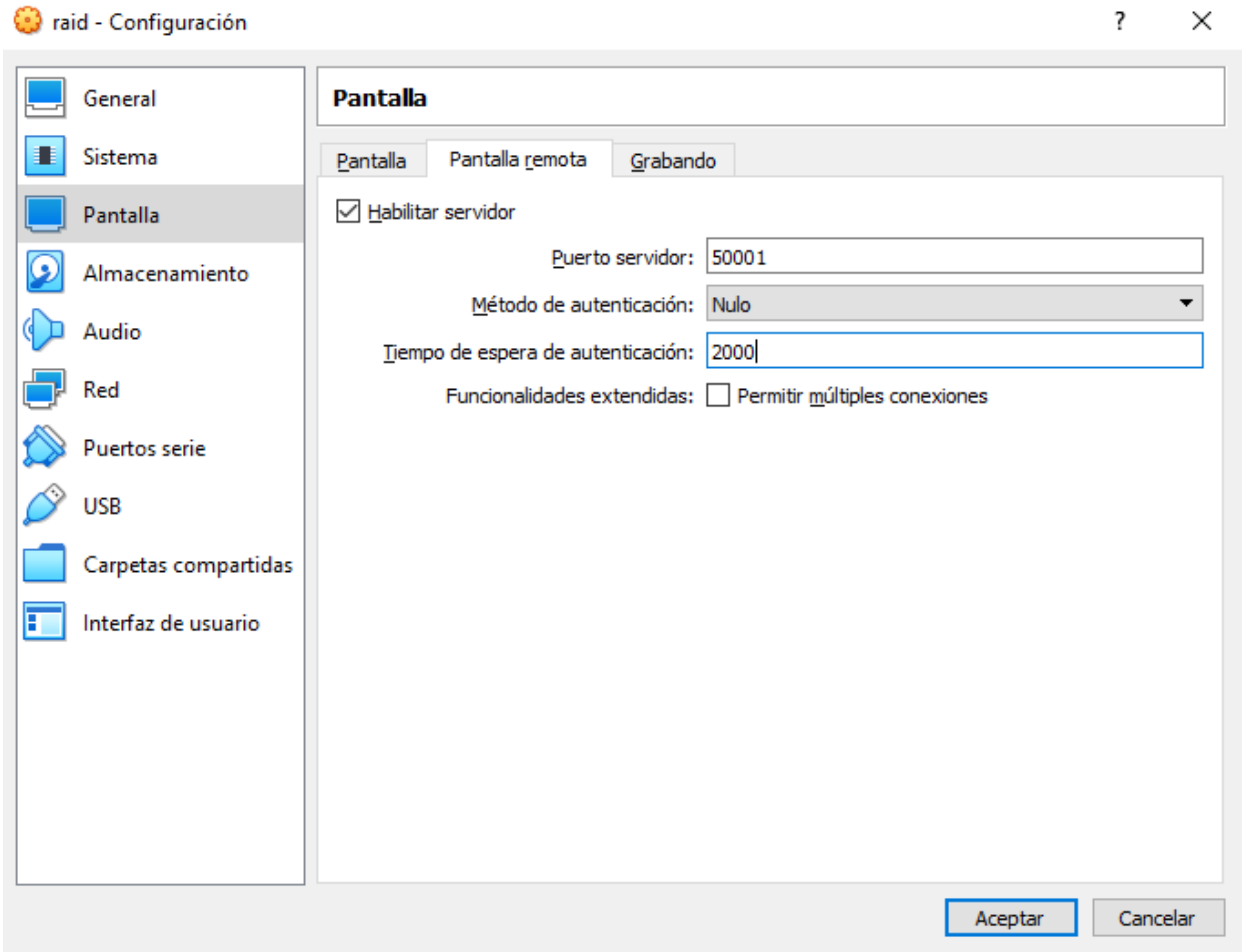

Ilustración H-0-2 Configuración RDP en VirtualBox

Fichero de configuración reducido para i3.

A continuación, detallará el contenido de un fichero de configuración para el gestor de ventana i3, mediante el cual no se mostrará nada que la aplicación que se ejecute sobre él.

```
  # i3 config file (v4)
  set $mod Mod4
  font pango:monospace 8
  bindsym $mod+Shift+q kill
  bindsym $mod+Shift+r restart
  bindsym $mod+Shift+e exec "i3-nagbar -t warning -m 'Presionate la
combianacion de tecla para salir de i3. ¿Esta seguro? Entonces cerrara
la sesion X.' -b 'Si,salir de i3' 'i3-msg exit'"
```

El fichero incluye una línea que especifica la versión del fichero de configuración, se especifica como tecla modificadora la tecla [Win] y como fuente monospace con tamaño 8 para los textos del gestor de ventana.

En este fichero de configuración también se detalla los siguientes atajos de teclados:

- [Win] + [Shift] + [q] que permite cerrar la ventana actual.
- [Win] + [Shift] + [r] que permite refrescar el gestor de ventana.
- [Win] + [Shift] + [q] que cerrará el gestor de ventana, antes solicitará una confirmación.

Para obtener más información sobre las posibilidades de configuración acceder a la página http://i3wm.org/docs/userguide.html

ANEXO I: MODIFICACIONES Y PROCEDIMIENTOS DE OPENGNSYS

En el anexo se detallará el procedimiento para realizar la instalación mediante script del cliente ogAgent en los diferentes sistemas operativos, las modificaciones sobre el código de la web de gestión de Opengnsys para añadir funcionalidades deseada y los ficheros de configuración o configuraciones necesarias.

Instalación de cliente ogAgent.

El cliente ogAgent se trata de una herramienta usada junto a Opengnsys y que nos permite conocer desde la web de gestión el estado actual del equipo o enviarle acciones a realizar una vez iniciado el sistema operativo.

Para ello se pueden realizar de dos maneras:

- Mediante la ejecución del script launchOgagentInstaller en cliente ogLive, este programará la instalación del cliente para el próximo inicio del administrador del sistema operativo. Sus argumentos son:

Ndisk	Establece el número del disco donde se localiza el sistema operativo sobre el que realizar la instalación.
Npart	Establece el numero de la partición donde se localiza el sistema operativo sobre el que realizar la instalación.
windowsadmin	Indica la contraseña del administrador del sistema y solo es necesario si el sistema operativo destino es Windows.

- Mediante el procedimiento detallado en la página del manual de Opengsys "OGAgent: agente OpenGnsys para sistema operativo"[122]. Para ello se han creado una serie de script para cada sistema operativo que permite realizar el procedimiento de manera automática. Esto script deben ser ejecutado como administrador o superusurio y en el mismo directorio debe encontrar el instalador pertinente, excepto para el caso de Windows anterior a la versión 1.1.1b que será necesario el binario de la herramienta sed. Estos scripts y el binario se proporcionará junto a la memoria.

  En la ejecución únicamente se solicitará la dirección IP del servidor de Opengnsys.

  Se comprobará que la dirección IP es accesible y se procederá a la instalación del cliente ogAgent y tras finalizar se modificará el archivo de configuración para cambiar la dirección IP contra la que actuó ogAgent por la dirección IP introducida al iniciar la ejecución.

  Si no ocurriera ningún error se terminará con la ejecución del script con la activación del servicio para que el cliente se autoinicie.

Modificación para crear menú para operadores

A continuación, se detallará que puntos del código fuente de la web de gestión han sido modificado con el fin de que pueda ser reproducido en otras versiones de Opengnsys.

- En el archivo includes/constantes.php se debe añadir el siguiente fragmento de código tras la definición de la variable ITEM_PRIVADO:

```
$ITEM_OPERADOR=0x0003;
```

- En el archivo principal/administracion.php se modifica la función SubarbolXML_superadministradores y la función CreacontextualXMLUsuarios.

  En la función SubarbolXML_superadministradores de debe añadir recuperación del valor global de la variable OPERADOR para ello se añadir el siguiente fragmento de código tras la definición de la variable ADMINISTRADOR:

```
global $OPERADOR;
```

Se debe modificar la sentencia donde se recupera los usuarios de la base de datos para incluir los usuarios de tipo operador, para ellos se debe sustituir la consulta:

```
SELECT idusuario,nombre,idtipousuario FROM usuarios WHERE
idtipousuario=".$SUPERADMINISTRADOR." OR
idtipousuario=".$ADMINISTRADOR." ORDER by idtipousuario,nombre
```

Por esta consulta:

```
SELECT idusuario,nombre,idtipousuario FROM usuarios WHERE
idtipousuario=".$SUPERADMINISTRADOR." OR
idtipousuario=".$ADMINISTRADOR." OR idtipousuario=".$OPERADOR." ORDER
by idtipousuario,nombre
```

Y, por último, se modifica el código encargado de mostrar cada usuario para mostrar una imagen distinta cuando se trate de un operador, para ellos se debe sustituir el código:

```
else
 $cadenaXML.=' imagenodo="../images/iconos/administradores.gif"';
```

Por este fragmento de código:

```
else if($rs->campos["idtipousuario"]==$ADMINISTRADOR)
 $cadenaXML.=' imagenodo="../images/iconos/administradores.gif"';
else if($rs->campos["idtipousuario"]==$OPERADOR)
 $cadenaXML.=' imagenodo="../images/iconos/operadores.gif"';
```

En la función CreacontextualXMLUsuarios se debe añadir recuperación del valor global de la variable OPERADOR para ello se añadir el siguiente fragmento de código tras la definición de la variable SUPERADMINISTRADOR:

```
global $OPERADOR;
```

Y se modifica el código encargado de mostrar las opciones de creación de usuario y de esta manera añadir el tipo de usuario operador a la lista, para ello debe añadir el siguiente fragmento al final de la función y antes de return:

```
// Crear operador
$wLeft=140;
$wTop=115;
$wWidth=400;
$wHeight=320;
```

```
  $wpages="../propiedades/propiedades_usuarios.php?idtipousuario=".$O
PERADOR;
  $wParam=$wLeft             .",".$wTop.",".$wWidth.",".$wHeight.",'".
$wpages."'";
  $layerXML.='<ITEM';
  $layerXML.=' alpulsar="insertar('.$wParam.',0,3)"';
  $layerXML.=' imgitem="../images/iconos/operadores.gif"';
  $layerXML.=' textoitem='.$TbMsg[14];
  $layerXML.='></ITEM>';
  $layerXML.='</MENUCONTEXTUAL>';
```

En caso de realizar la instalación también habría que modificar principal/administración.device.php de la misma manera.

- En el archivo varios/acceso_operador.php se modifica el uso de la variable de sesión swop por el uso de la variable de sesión swoptipo. Para ello se sustituye el fragmento de código:

```
  $ITEMS_PUBLICOS=1;
  $ITEMS_PRIVADOS=2;
  if (isset($_SESSION["swop"])){
  // Acceso al menu de adminitración del aula
  $wurl="menucliente.php?tip=".$ITEMS_PRIVADOS;
  Header('Location:'.$wurl);
  }
```

Por este fragmento:

```
  $ITEMS_PUBLICOS=1;
  $ITEMS_PRIVADOS=2;
  $ITEMS_OPERADOR=3;

  if (isset($_SESSION["swoptipo"])){
   if($_SESSION["swoptipo"]==1){
   // Acceso al menu de adminitración del aula
   $wurl="menucliente.php?tip=".$ITEMS_PRIVADOS;
   }
   // Acceso al menu de operador del aula
   if($_SESSION["swoptipo"]==2){
    $wurl="menucliente.php?tip=".$ITEMS_OPERADOR;
   }
   Header('Location:'.$wurl);
  }
```

- En el archivo varios/accesoperadores.php se modifica sustituye el código tras la consulta del usuario que se autentificado para que en función su tipo se inicialicen las variables tip, que indica el tipo de elementos a mostrar, y swoptipo, que indica el tipo de usuario autentificado. Para ello se sustituye el fragmento de código:

```
  $wurl="menucliente.php?tip=".$ITEMS_PRIVADOS;
  $_SESSION["swop"]=$usu;
  Header('Location:'.$wurl);
  exit;
```

Por este fragmento:

```
  $idtipousuario=$rs->campos["idtipousuario"];

  global $ITEM_PUBLICO;
  global $ITEM_PRIVADO;
  global $ITEM_OPERADOR;
  global $SUPERADMINISTRADOR;
  global $ADMINISTRADOR;
  global $OPERADOR;
  $wurl="menucliente.php?tip=".$ITEM_PUBLICO;
  if($idtipousuario==$SUPERADMINISTRADOR                                    ||
$idtipousuario==$ADMINISTRADOR ){
   $wurl="menucliente.php?tip=".$ITEM_PRIVADO;
   $_SESSION["swoptipo"]=$ITEM_PRIVADOS;
  }

  if($idtipousuario==$OPERADOR){
   $wurl="menucliente.php?tip=".$ITEM_OPERADOR;
   $_SESSION["swoptipo"]=$ITEM_OPERADOR;
  }

  $_SESSION["swop"]=$usu;
  Header('Location:'.$wurl);
  exit;
```

- En el archivo varios/accionmenu.php se modifica la función pintaMenus para añadir la opción de seleccionar como tipo de elemento el tipo operador. se debe añadir recuperación del valor global de la variable ITEM_OPERADOR para ello se añadir el siguiente fragmento de código tras la definición de la variable ITEM_PRIVADO:

```
  global $OPERADOR;
```

Y se modificar la inclusión de las opciones en el desplegable de tipo de elemento para añadir la opción de operador. Para ello se debe sustituir el fragmento de código:

```
  $parametros.=$ITEM_PRIVADO."=".$TbMsg[6]."";
```

Por el siguiente fragmento:

```
  $parametros.=$ITEM_PRIVADO."=".$TbMsg[6]."".chr(13);
  $parametros.=$ITEM_OPERADOR."=".$TbMsg[15]."";
```

- En el archivo varios/informacion_menus.php se modifica la función SubarbolXML_Items para mostrar las acciones de tipo operador que posee un menú concreto. Para ello se debe sustituir el contenido de esta función por el siguiente código:

```
  function SubarbolXML_Items($cmd,$idmenu){
   global  $TbMsg;
   global  $ITEM_PUBLICO;
   global  $ITEM_PRIVADO;
   global  $ITEM_OPERADOR;
   global  $idcentro;
   global  $EJECUCION_PROCEDIMIENTO;
   global  $EJECUCION_TAREA;
```

```
 global  $contitempub;
 global  $contitempri;

 $cadenaXML="";
 $rs=new Recordset;
 $cmd->texto="SELECT acciones_menus.*,iconos.urlicono as urlimg
 FROM  acciones_menus
 LEFT OUTER JOIN iconos ON acciones_menus.idurlimg =iconos.idicono
 WHERE acciones_menus.idmenu=".$idmenu."
  ORDER BY acciones_menus.tipoitem,acciones_menus.orden";
$rs->Comando=&$cmd;

if (!$rs->Abrir()) return($cadenaXML); // Error al abrir recordset
$rs->Primero();

$tbmodalidad[1]=$TbMsg[18];
$tbmodalidad[2]=$TbMsg[19];
$swpub=false;
$swpriv=false;
$swoper=false;

$cadenaXML.='<ITEMS';
$cadenaXML.=' imagenodo="../images/iconos/carpeta.gif"';
$cadenaXML.=' infonodo="'.$TbMsg[22].'"';
$cadenaXML.='>';

while (!$rs->EOF){
 if ($rs->campos["tipoitem"]==$ITEM_PUBLICO){
 $contitempub++;
  if (!$swpub) {
   $cadenaXML.='<ITEMSPUBLICOS';
   $cadenaXML.=' imagenodo="../images/iconos/carpeta.gif"';
   $cadenaXML.=' infonodo="'.$TbMsg[13].'"';
   $cadenaXML.='>';
   $swpub=true;
  }
 }
 if ($rs->campos["tipoitem"]==$ITEM_PRIVADO){
  $contitempri++;
  if ($swpub) {
   $cadenaXML.='</ITEMSPUBLICOS>';
   $swpub=false;
  }
  if (!$swpriv) {
   $cadenaXML.='<ITEMSPRIVADOS';
   $cadenaXML.=' imagenodo="../images/iconos/carpeta.gif"';
   $cadenaXML.=' infonodo="'.$TbMsg[14].'"';
   $cadenaXML.='>';
   $swpriv=true;
  }
 }
 if ($rs->campos["tipoitem"]==$ITEM_OPERADOR){
  $contitempri++;
  if ($swpub) {
   $cadenaXML.='</ITEMSPUBLICOS>';
   $swpub=false;
```

```
    }
    if ($swpriv) {
     $cadenaXML.='</ITEMSPRIVADOS>';
     $swpriv=false;
    if (!$swoper) {
     $cadenaXML.='<ITEMSRECUPERACION';
     $cadenaXML.=' imagenodo="../images/iconos/carpeta.gif"';
     $cadenaXML.=' infonodo="'.$TbMsg[23].'"';
     $cadenaXML.='>';
     $swoper=true;
    }

  switch($rs->campos["tipoaccion"]){
     case $EJECUCION_PROCEDIMIENTO :
      $cmd->texto='SELECT       procedimientos.descripcion       FROM
procedimientos      WHERE      procedimientos.idprocedimiento='.$rs-
>campos["idtipoaccion"];
      $urlimg="procedimiento.gif";
      break;
     case $EJECUCION_TAREA :
      $cmd->texto='SELECT    tareas.idtarea,  tareas.descripcion  FROM
tareas WHERE tareas.idtarea='.$rs->campos["idtipoaccion"];
      $urlimg="tareas.gif";
      break;
   }
   if(!empty($rs->campos["idtipoaccion"]))
     $cadenaXML.= SubarbolXML_itemsmenus($cmd,$urlimg,$rs->campos);
     $rs->Siguiente();
   }

  if ($swpub)
     $cadenaXML.='</ITEMSPUBLICOS>';
  if ($swpriv)
     $cadenaXML.='</ITEMSPRIVADOS>';
  if ($swoper)
     $cadenaXML.='</ITEMSRECUPERACION>';

 $cadenaXML.='</ITEMS>';
 $rs->Cerrar();
  return($cadenaXML);
}
```

- En el archivo varios/menucliente.php se modifica para añadir una nueva opción en el switch que tiene como argumento tip (que identifica el tipo de elementos a mostrar) que existe tras la llamada a la función RecuperaMenu esta nueva cláusula contiene el siguiente código:

```
  case $ITEMS_OPERADOR:
  if(!empty($rsmenu->campos["htmlmenupri"])){
   $urlHtml=$rsmenu->campos["htmlmenupri"];
   $urlHtml="http://".$urlHtml;
   Header('Location: '.$urlHtml);
  } else {
   $_SESSION["widcentro"]=$rsmenu->campos["idcentro"];
   $codeHtml=GeneraMenu($rsmenu,$ITEMS_OPERADOR,$iph);
  }
```

```
  break;
```

Para que esta funcione correctamente es necesario definir el valor para ITEMS_OPERADOR y para ello se añade el siguiente fragmento de código tras la definición de la variable ITEMS_PRIVADOS:

```
  $ITEMS_OPERADOR=3;
```

- En el archivo controlpostacceso.php se modifica las consultas en la función toma_datos para permitir recuperar usuarios de tipo operador y de esta manera autentificar correctamente con este tipo de usuario para ello se elimina de ambas consultas el filtro de que el tipo de usuario se distinto de 3.

Para limitar el acceso de los operadores únicamente para cuando acedan desde el cliente ogLive se la cláusula que existe tras la llamada a toma_datos y que indica que no ha podido recuperarse un usuario valida, esta pasa de la siguiente condición:

```
  if(!$resul)
```

A esta condición:

```
  $OPERADOR=3;
   if(!$resul || (empty($iph) && $tsu == $OPERADOR))
```

Y además se modifica toma_datos el fuerce para que tanto administrador como superadministrador accedan como administrador de Unidad organizativa. Para ello se sustituye el siguiente fragmento de código:

```
  $idtipousuario=2; // Fuerza al acceso como administrador de Unidad
organizativa
```

Por este fragmento de código:

```
  if($idtipousuario == 1){
    $idtipousuario=2; // Fuerza al acceso como administrador de Unidad
organizativa cuando sea superadministrador y se seleccione un centro
    }
```

- En el archivo validacion/functions.php se modifica la función RecuperaMenu para imprimir un texto que indique que se ha autentificado como operador o como administrador para ello se añade el siguiente fragmento tras la declaración del título del menú:

```
  if($tipo==$ITEMS_PRIVADOS)
    $codeHTML.='<SPAN   style="COLOR:   #999999;FONT-FAMILY:   Arial,
Helvetica, sans-serif;FONT-SIZE: 20px;"><U>'.$TbMsg[5].'</U></SPAN>';
  if($tipo==$ITEMS_OPERADOR)
    $codeHTML.='<SPAN   style="COLOR:   #999999;FONT-FAMILY:   Arial,
Helvetica, sans-serif;FONT-SIZE: 20px;"><U>'.$TbMsg[6].'</U></SPAN>';
```

Se modifica la condición para seleccionar el número de columnas y de esta manera se use el mismo número de columnas para elementos privados y de operador. Para ello se sustituye el siguiente fragmento de código:

```
  if($tipo==$ITEMS_PRIVADOS)
   $mod=$smodalidad;
  else
   $mod=$modalidad;
```

Por este fragmento de código:

```
if($tipo==$ITEMS_PRIVADOS || $tipo==$ITEMS_OPERADOR)
 $mod=$smodalidad;
else
 $mod=$modalidad;
```

Se modifica para que se añada el botón para volver al menú de elementos público desde el menú de operador. Para ello se sustituye el siguiente fragmento de código:

```
if (empty($url)) $url="";
 switch($tipo){
  case $ITEMS_PUBLICOS:
   $url.='acceso_operador.php';
   $lit="Administrar";
   break;
  case $ITEMS_PRIVADOS:
   $url.='menucliente.php';
   $lit="Volver";
  break;
}
```

Por este fragmento de código:

```
if (empty($url)) $url="";
 switch($tipo){
  case $ITEMS_PUBLICOS:
   $url.='acceso_operador.php';
   $lit="Administrar";
   break;
  case $ITEMS_OPERADOR:
  case $ITEMS_PRIVADOS:
   $url.='menucliente.php';
   $lit="Volver";
  break;
}
```

Para el funcionamiento de esta modificaciones es necesario recuperar el valor de las variables ITEMS_OPERADOR y TbMsg (variable donde se almacenan texto con un idioma determinado). Para ello se debe añadir al inicio de la función el siguiente fragmento:

```
global $ITEMS_OPERADOR;
global $TbMsg;
```

Para terminar y en el caso de que se quiera que en el menú de administrador se muestre tanto los elementos de tipo privado como elementos de operador será necesario modificar la siguiente condición para indicar que en caso de estar recuperando el menú con elementos privados también se recupere los elementos de operador:

```
if($tipoitem==$tipo)
```

Dejando la condición de esta manera:

```
if($tipoitem==$tipo ||
   ($tipo==$ITEMS_PRIVADOS && $tipoitem==$ITEMS_OPERADOR))
```

- Para los archivos de idioma se detallará los elementos incluido para el idioma español, ya que para el resto son extrapolable de manera sencilla.

En el fichero idiomas/php/esp/accionmenu_esp.php se añade el siguiente fragmento al final del archivo:

```
$TbMsg[15]='Operador';
```

En el fichero idiomas/php/esp/administracion_esp.php se añade el siguiente fragmento al final del archivo:

```
$TbMsg[14]='"Nuevo Operador"';
```

En el fichero idiomas/php/esp/informacion_menus_esp.php se añade el siguiente fragmento al final del archivo:

```
$TbMsg[23]='Items operador';
```

En el fichero idiomas/php/esp/menucliente_esp.php se añade el siguiente fragmento al final del archivo:

```
$TbMsg[5]='Men&uacute; de administrador';
$TbMsg[6]='Men&uacute; de operador';
```

Modificación de la barra superior de la web de gestión de Opengnsys

A continuación, se detallará que puntos del código fuente de la web de gestión han sido modificado con el fin de que pueda ser reproducido en otras versiones de Opengnsys.

- El archivo barramenu.php se modifica para añadir en las funciones definida en el archivo personalizado/modificador_menu.php para ello se añade el siguiente fragmento al comienzo del archivo:

```
include_once("./personalizado/modificador_menu.php");
```

Y luego es necesario añadir la llamada a la función añadirOpcionGestionAvanzada justo después del último elemento que se creará en la barra de menú para ello se debe colocar tras el siguiente fragmento de código:

```
 <TD     onclick=eleccion(this,21);     onmouseout=desresaltar(this);
onmouseover=resaltar(this); align=middle>
  <A   href="#"   style="text-decoration:   none"><IMG   border=0
src="./images/iconos/usuarioslog.gif"> <SPAN      class=menupral
><?php echo  $TbMsg[10] ?></SPAN></A> </TD>
 <TD                    width=4                    align=middle><IMG
src="./images/iconos/separitem.gif"></TD>
```

Se debe incluir el siguiente fragmento de código que realizará la llamada:

```
<?php
 añadirOpcionGestionAvanzada($TbMsg);
?>
```

- Para los archivos de idioma se detallará los elementos incluido para el idioma español, ya que para el resto son extrapolable de manera sencilla. En el fichero idiomas/php/esp/barramenu_esp.php se añade el siguiente fragmento al final del archivo:

```
$TbMsg[17]='Opciones Avanzadas';
$TbMsg[18]='Guacamole';
$TbMsg[19]='Sincronizaci&oacute;n con Guacamole';
```

```
  $TbMsg[20]='Gestor de ficheros';
  $TbMsg[21]='Gesion de switch';
```

Modificación de la página aula de la web de gestión de Opengnsys

A continuación, se detallará que puntos del código fuente de la web de gestión han sido modificado con el fin de que pueda ser reproducido en otras versiones de Opengnsys.

- El archivo aula.php se modifica para añadir en las funciones definida en el archivo personalizado/ modificador_aula.php para ello se añade el siguiente fragmento al comienzo del archivo:

```
  include_once("../personalizado/modificador_aula.php");
```

Se incluye la etiqueta HEAD la llamada para incluir las dependencias que se realiza con el siguiente fragmento de código:

```
  <?php
   añadirDependencias();
  ?>
```

Las modificaciones en la función pintaordenadores son las siguientes:

  - Se recupera el valor de la variable nombreambito que especificará el nombre del aula o del subgrupo de esta en caso de encontrarse en él. Para ello se añade el siguiente fragmento tras la declaración de la variable nombreaula:

```
  global $nombreambito;
```

  - Se realiza la llamada a la función añadirInicioModificacion. Para ello se añade el siguiente fragmento tras creación de etiqueta con la mac del equipo:

```
  añadirInicioModificacion($nombreambito);
```

  - Se realiza la llamada a la función añadirOpcionesModificacion. Para ello se añade el siguiente fragmento tras la declaración de todas las variables:

```
  añadirOpcionesModificacion($Mnombreordenador[$i],$Mmac[$i],$Mip[$i]
,$TbMsg);
```

  - Se realiza la llamada a la función añadirFinModificacion. Para ello se añade el siguiente fragmento tras el cierre de la tabla:

```
  añadirFinModificacion($nombreambito,$TbMsg);
```

- Para los archivos de idioma se detallará los elementos incluido para el idioma español, ya que para el resto son extrapolable de manera sencilla.

En el fichero idiomas/javascripts/xxx/aulas_xxx.js se añade el siguiente fragmento al final del archivo:

```
  TbMsg[13]="Pantalla de entrada de comando";
  TbMsg[14]="Intoduzca el comando a enviar";
  TbMsg[15]="El comando a ejecutar no hara uso del Shebang
indicado.<br>Para Windows hay que escapar las contrabarras(\\\\).";
  TbMsg[16]="Enviar Comado";
  TbMsg[17]="Contiene un Shebang que del cual no se harán uso";
  TbMsg[18]="No se escribio ningun comando";
```

```
  TbMsg[19]="Los equipos a apagar son ";
  TbMsg[20]="Los equipos a encender son ";
  TbMsg[21]="Los equipos a enviar archivos son ";
  TbMsg[22]="Los equipos a enviar comando son ";
  TbMsg[23]="Los equipos a enviar procedimiento son";
  TbMsg[24]="Se debe seleccionar un procedimiento a enviar";
  TbMsg[25]="No se ha seleccionado ningun sistema operativo";
  TbMsg[26]="Sistemas Operativos a enviar comandos";
  TbMsg[27]="Ha ocurriodo un error";
```

En el fichero idiomas/php/xxx/aulas_xxx.php se añade el siguiente fragmento al final del archivo:

```
  $TbMsg[51]='"Enviar operaciones"';
  $TbMsg[52]='"Borrar selección"';
  $TbMsg[53]='Enceder';
  $TbMsg[54]='Apagar';
  $TbMsg[55]='Enviar Archivos';
  $TbMsg[56]='Enviar Comando';
  $TbMsg[57]='Enviar procedimiento';
  $TbMsg[58]='Ninguna';
  $TbMsg[59]='Seleccionar acción para todos';
  $TbMsg[60]='Ventana seleccion de ficheros a enviar y procedimiento a
usar';
  $TbMsg[61]='Mostrar/Ocultar <br>seleccion operaciones';
  $TbMsg[62]='Procedimiento a enviar';
  $TbMsg[63]='Ninguno';
```

Fichero de configuración configuracion_sincronizacion_guacamole.php

Se detallará el contenido del fichero configuracion_sincronizacion_guacamole.php que será el usado para indicar los parámetros usados en la sincronización entre Guacamole y Opengnsys.

El contenido del fichero será:

```
  <?php
  $conexion_guacamole = array(
      "url" => "localhost:3306",
      "username" => "guacamole",
      "password" => "some_password",
      "database" => "guacamole_db",
      "supplier" => "mysql",
   );

  $cadenaconexion_guacamole=implode(";", $conexion_guacamole);

  $nombreAdministrador = "guacadmin";
  $subfijo_vnc = "_VNC";
  $subfijo_ssh = "_SSH";
  $subfijo_rdp = "_RDP";

  ##################  CONFIGURACION   VNC  #############
  $vnc_pass = "VNCPASS";
  $vnc_port = "5900";
```

```
  ################  CONFIGURACION   SSH  ############
  $ssh_config = "pass"; // pass privateKey privateKey&passphrase
  $ssh_user = "username";
  $ssh_pass = "password";
  $ssh_port = "22";
  $ssh_private = "private-key";
  $ssh_passphrase = "passphrase";
  ################  CONFIGURACION   RDP  ############
  $rdp_user = "username";
  $rdp_pass = "password";
  $rdp_port = "3389";
  $rdp_ignore_cert = "true"; // Para permitir la conexion sin valida
el certificado del PC al que conectarno
  ?>
```

Los fragmentos en lo que se divide el fichero son:

- La variable conexion_guacmole define los parámetros para realizar la conexión a la base datos de Guacamole. Los parámetros son:

url	Indica el ubicación y puerto donde se ubica la base datos.
username	Indica el nombre del usuario que se usará para acceder a la base de datos.
password	Indica la contraseña del usuario que se usará para acceder a la base de datos.
database	Indica el nombre por el cual se identifica la base de datos de Guacamole.
supplier	Indica el proveedor de la base de datos. Actualmente únicamente esta soportado mysql.

- La variable cadenaconexion_guacamole será usada para establecer la conexión con la base datos mediante el uso de la función CreaComando definida en Opengnsys.
- La variable nombreAdministrador indicará el usuario de Guacamole al cual se le asignará las conexiones creada o actualizadas.
- Las variables subfijo_X especificará el sufijo que junto al nombre del equipo identificará la conexión para cada protocolo.
- Las variables usadas para la configuración mediante el protocolo VNC serán:

vnc_pass	Indica la contraseña de acceso al servidor VNC.
vnc_port	Indica el puerto para la conexión VNC.

- Las variables usadas para la configuración mediante el protocolo SSH serán:

ssh_config	Indica el método de autentificación que se usará para establecer la conexión SSH. Los posibles valores son: pass, privateKey o privateKey&passphrase
ssh_user	Indica el nombre del usuario para la autentificación de la conexión SSH.
ssh_pass	Indica la contraseña del usuario para la autentificación de la conexión SSH si el valor de ssh_config fuera pass.

ssh_port	Indica el puerto de conexión contra el servidor SSH.
ssh_private	Indica la clave privada completa que se usará para la autentificación de la conexión SSH si el valor de ssh_config fuera privateKey o privateKey&passphrase.
ssh_passphrase	Indica contraseña de la clave privada indica en ssh_private y que se usará para la autentificación de la conexión SSH si el valor de ssh_config fuera privateKey&passphrase.

- Las variables usadas para la configuración mediante el protocolo RDP serán:

rdp_user	Indica el nombre del usuario para la autentificación de la conexión RDP.
rdp_pass	Indica la contraseña del usuario para la autentificación de la conexión RDP.
rdp_port	Indica el puerto de conexión contra el servidor RDP.
rdp_ignore_cert	Indica si se quiere ignorar la validación del certificado a la hora de realizar la conexión, esto es necesario cuando el servidor RDP es quien genera su propio certificado.

Fichero de configuración configuracion_fichero.php

Se detallará el contenido del fichero configuracion_fichero.php tanto para la configuración que será usado tanto el gestor de ficheros como a la hora de enviar archivos al cliente ogLive de forma masiva desde la web de gestión de Opengnsys.

El contenido del fichero será:

```
  <?php
   $directorio_relativo_ficheros = "ficheros/";
   $directorio_completo_en_ogLive = "/opt/opengnsys/images/groups/";
   $directorio_absoluto_ficheros                                    =
"/opt/opengnsys/www/personalizado/ficheros/";
   $fichero_script_copia = "copiaRemotaScript";
   $fichero_ultima_seleccion = "copiaRemotaListado";
   $comandoParaCopia          =          "#Ejecutar          Copia\ncp
$directorio_completo_en_ogLive$fichero_script_copia  /tmp/\nchmod  700
/tmp/$fichero_script_copia\n/tmp/$fichero_script_copia";
  ?>
```

Los fragmentos en lo que se divide el fichero son:

- La variable directorio_relativo_ficheros será usada para indicar la ruta desde el directorio /opt/opengnsys/www/personalizado de forma masiva. Este indicará un enlace simbólico ya que permitirá reducir los directorios que son en la cual se alojarán los ficheros que se administrarán con el gestor de fichero y que podrán ser enviado al cliente ogLive mostrado al exterior.
- La variable directorio_completo_en_ogLive será usada indicar la ruta en el cliente ogLive donde se montará mediante samba el directorio del servidor indicado en la variable directorio_absoluto_ficheros.
- La variable directorio_absoluto_ficheros indica la ruta absoluta a la que apuntará el enlace simbólico al que hace referencia la variable directorio_relativo_ficheros.
- La variable fichero_script_copia será usada indicar el nombre del script que se usará en el proceso de

envió de archivos a los clientes ogLive.

- La variable fichero_ultima_seleccion será usada para almacenar los últimos archivos y directorios que se seleccionaron para realizar él envió de archivo a los clientes ogLive.
- La variable comandoParaCopia será usada indica el script que será enviado mediante la acción Enviar Script desde la web de gestión de Opengnsys. El contenido del script enviado será:

```
#Ejecutar Copia
cp $directorio_completo_en_ogLive$fichero_script_copia /tmp/
chmod 700 /tmp/$fichero_script_copia
/tmp/$fichero_script_copia
```

Este script copiará el script indicado en la variable fichero_script_copia, se configurará para poder ser ejecutado y finalmente ejecutará el script el cual realizará la copia de fichero.

Fichero de configuración mapeo_particiones_opengnsys.php

Se detallará el contenido del fichero mapeo_particiones_opengnsys.php que será el usado para indicar el contenido de las diferentes particiones que tendrán los equipos del laboratorio de telemática.

```
<?php
$mapeoParticiones = array(
"Windows" =>   array(
  "particion" => "/mnt/sda1",
  "comandoComprobacion"   => 'ver | findstr /C:\"Versi\"| findstr
/C:\"n 10.\" > nul 2> nul & if NOT errorlevel  1 ( *comando* )' ,
  "comandoComentario" => "rem ",
  "lugarComentario" => "fin",
  "esSistemaOperativo" => true,
  ),
 "Linux CentOS" =>  array(
  "particion" => "/mnt/sda2",
  "comandoComprobacion"  => 'if lsb_release -d | grep -q \"CentOS
Linux release 7\"; then \n*comando*\nfi' ,
  "comandoComentario" => "# ",
  "lugarComentario" => "principio",
  "esSistemaOperativo" => true,
  ),
"Linux Ubuntu/Debian" =>   array(
  "particion" => "/mnt/sda6",
  "comandoComprobacion"  => 'if lsb_release -d | grep -q \"Ubuntu
16\"; then \n*comando*\nfi' ,
  "comandoComentario" => "# ",
  "lugarComentario" => "principio",
  "esSistemaOperativo" => true,
  ),
 "Linux programas" =>   array(
  "particion" => "/mnt/sda7",
  "comandoComprobacion"  => "" ,
  "comandoComentario" => "",
  "lugarComentario" => "",
  "esSistemaOperativo" => false,
  ),
 "Windows Programas" =>   array(
  "particion" => "/mnt/sda8",
```

```
    "comandoComprobacion"  => "" ,
    "comandoComentario" => "",
    "lugarComentario" => "",
    "esSistemaOperativo" => false,
    ),
  "Linux Moviles" =>   array(
    "particion" => "/mnt/sda9",
    "comandoComprobacion"  => 'if lsb_release -d | grep -q \"Debian
GNU/Linux Kali Linux 1.0\"; then \n*comando*\nfi' ,
    "comandoComentario" => "# ",
    "lugarComentario" => "principio",
    "esSistemaOperativo" => true,
    ),
   "Linux Kali" =>   array(
    "particion" => "/mnt/sda10",
    "comandoComprobacion"  => 'if lsb_release -d | grep -q\"Debian
GNU/Linux Kali Linux 1.0\"; then \n*comando*\nfi' ,
    "comandoComentario" => "# ",
    "lugarComentario" => "principio",
    "esSistemaOperativo" => true,
  ),
);
?>
```

Por cada partición será necesario añadir la siguiente estructura:

```
  "nombre" =>   array(
    "particion" => "/mnt/sdXY",
    "comandoComprobacion" => '…*comando*…' ,
    "comandoComentario" => "…",
    "lugarComentario" => "principio",
    "esSistemaOperativo" => true,
  )
```

Los partes en la que se divide la estructura es:

- nombre que se establece como identificar la partición.
- particion que establece el punto de montaje para la partición en el cliente ogLive.
- comandoComprobacion contendrá una cláusula que permitirá solo ejecutar un script si se cumple la comprobación. Esto será usado comprobar se ejecutando un sistema operativo determinado. Es necesario incluir la palabra clave *comando*, la cual será usado para identificar el lugar donde debe ubicarse el comando o script a ejecutar.
- comandoComentario contendrá el método necesario para establecer un comentario al ejecutar un script en un sistema operativo determinado.
- lugarComentario indica donde se debe colocará el comentario al ejecutar comando desde la modificación de aula. El valor se esta variable solo puede de ser "principio" o "fin".
- esSistemaOperativo indica que la partición que se definida en la estructura contendrá un sistema operativo y será usado para mostrar como posibles opciones a la hora de enviar comandos desde la modificación de aula. El valor se esta variable solo puede de ser true o false.

Solo será necesario rellenar las variables comandoComprobacion, comandoComentario y lugarComentario si se especifica la variable esSistemaOperativo como true.

Configuración recomendada para gestor_fichero.php

A continuación, se detallará un conjunto de configuraciones necesarias para restringir el acceso a los ficheros del servidor y aumentar el límite en el tamaño de los archivos que se permitirán subir.

Para limitar el acceso a los ficheros del servidor se deberán seguir los siguientes pasos como superusuario:

1. Crear un enlace simbólico a la carpeta la cual montará en el cliente ogLive. Para ello se usará la herramienta ln[123].

```
 ln                -fs                /opt/opengnsys/images/groups/
/opt/opengnsys/www/personalizado/ficheros/
```

Los argumentos usados son:

-f	Indica que si existiera ya el enlace en el destino se borre y se cree de nuevo.
-s	Indica que se cree un enlace simbólico.
/opt/opengnsys/images/groups/	Indica la ruta de origen para la cual se creará el enlace.
/opt/opengnsys/www/personalizado/ficheros/	Indica destino donde crear el enlace.

2. Modificar las variables del fichero de configuración configuracion_fichero.php para que su valor sea el siguiente:

```
 $directorio_relativo_ficheros = "ficheros/";
 $directorio_completo_en_ogLive = "/opt/opengnsys/images/groups/";
 $directorio_absoluto_ficheros                                    =
"/opt/opengnsys/www/personalizado/ficheros/";
```

3. Cambiar el propietario para que sea posible el acceso al directorio por el usuario www-data. Para ello se hará uso de la herramienta chown.

```
 chown www-data -R /opt/opengnsys/www/personalizado/ficheros/
```

Los argumentos usados son:

www-data	Indica el nuevo propietario de los ficheros.
-R	Indica que el cambio de propietario se haga de manera recursiva.
/opt/opengnsys/www/personalizado/ficheros/	Indica el directorio al cual se cambiará de propietario.

4. Crear un fichero .htaccess para limitar el acceso a los ficheros a no permitir que el directorio sea navegable. Para ello se creará el fichero /opt/opengnsys/www/personalizado/ficheros/.htaccess con el siguiente contenido:

```
 Options -Indexes
```

Para aumentar el límite en el tamaño de los archivos que se permitirán subir se debe modificar en el fichero de configuración /etc/php/X/fpm/php.ini (siendo X la version de php que este instalada) los siguientes valores:

post_max_size	Indica el tamaño máximo de datos de POST permitidos. Cuando se usa un solo un número, el valor de este es medido en bytes. También indica la unidad de medida mediante uso de letra.
upload_max_filesize	Indica el tamaño máximo de un fichero subido, este debe ser menor a post_max_size, ya que en los datos POST pueden subir más de un fichero. El método de especificar el valor es el mismo que para post_max_size.
max_file_uploads	Indica el número máximo de fichero que es posible subir en una solo petición.

ANEXO J: CONTROL REMOTO MEDIANTE GUACAMOLE

El anexo se detallará el proceso de configuración de los diferentes servidores que será usado con guacamole para los protocolos VNC, RDP y SSH.

Servidor VNC en Linux

Como servidor VNC en sistema Linux se hará uso del servidor x11vnc junto un servicio autoiniciar el servicio, ya sea al inicio del sistema o porque el servidor VNC se cierre al cerrar una sesión.

Para realizar la configuración se ha de seguir los siguientes pasos como superusuario:

1. Instalar el paquete x11vnc para ello dependiendo de sistema operativo será necesario usar yum o apt.

```
#Para sistemas Debian, Ubuntu y derivados
apt install -y x11vnc
#Para sistemas Red Hat y derivados
yum install -y x11vnc
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
x11vnc	Es el nombre del paquete a instalar.

2. Crear el fichero de contraseña para el servidor VNC. Para ello se hará uso de la herramienta x11vnc.

```
x11vnc -storepasswd /opt/passwdVNC
```

Los argumentos usados serán:

-storepasswd	Indica se va a crear un nuevo fichero con la contraseña de acceso.
/opt/passwdVNC	Indica fichero a crear con la contraseña de acceso

Tras ejecutar el comando se pedirá introducir dos veces la contraseña y confirmar si se quiere crear el fichero.

3. Crear el servicio de que iniciará el servidor VNC. Para ello se debe crear el fichero /lib/systemd/system/x11vnc.service con el siguiente contenido:

```
[Unit]
Description=Start x11vnc at startup.
After=multi-user.target

[Service]
```

```
  Type=simple
  ExecStart=/usr/bin/x11vnc -auth guess -forever -loop -noxdamage -
rfbauth /opt/passwdVNC -display :0 -rfbport 5900 -shared
  Restart=always
  [Install]
  WantedBy=multi-user.target
```

Los argumentos usados serán con la herramienta x11vnc son:

-auth guess	Indica que se intente usar el fichero XAUTHORITY que este actualmente en uso para la conexión VNC
-forever	Indica que se mantenga escuchando siempre, aunque los clientes se desconecten.
-noxdamage	Deshabilita el uso de framebuffer, que evitar él envió de la pantalla completa.
-rfbauth /opt/passwdVNC	Establece el fichero que contiene la contraseña de acceso.
-display :0	Indica que se establezca la conexión contra el servidor de pantalla principal. Pudiendo compartir el mismo escritorio con el usuario físico.
-rfbport 5900	Indica el puerto usado para el servidor VNC.
-shared	Indica que la pantalla VNC será compartida por las diferentes conexiones.

4. Establecer los permisos adecuado al fichero generado para ello se hará uso de la herramienta chmod.

```
  chmod 644 /lib/systemd/system/x11vnc.service
```

Los argumentos usados serán:

644	Estable que los permisos para que todos los usuarios puedan leer el fichero y que solo el propietario pueda modificarlo.
/lib/systemd/system/x11vnc.service	Indica el fichero al cual se cambiará los permisos

5. Establecer el grupo y usuario propietario adecuado al fichero generado para ello se hará uso de la herramienta chown.

```
  chown root:root /lib/systemd/system/x11vnc.service
```

Los argumentos usados serán:

root:root	Estable que tanto el usuario propietario como el grupo sea root. El primero indica el usuario y el segundo indica el grupo.
/lib/systemd/system/x11vnc.service	Indica el fichero al cual se cambiará los permisos

6. Habilitar el servicio para que se inicie automáticamente al iniciar el sistema para ello se hará uso de la herramienta systemctl.

```
systemctl enable /lib/systemd/system/x11vnc.service
```

Los argumentos usados serán:

enable	Establece que el servicio que inicie con el arranque del sistema.
/lib/systemd/system/x11vnc.service	Indica el servicio que se iniciará automáticamente.

7. Iniciar el servicio mediante el uso de la herramienta systemctl.

```
systemctl start /lib/systemd/system/x11vnc.service
```

Los argumentos usados serán:

start	Indica que se incide un servicio.
/lib/systemd/system/x11vnc.service	Indica el servicio que se iniciará.

Servidor SSH en Linux

Para realizar la configuración se ha de seguir los siguientes pasos como superusuario:

1. Instalar el paquete ssh para ello dependiendo de sistema operativo será necesario usar yum o apt.

```
#Para sistemas Debian, Ubuntu y derivados
apt install -y openssh-server
#Para sistemas Red Hat y derivados
yum install -y openssh-server
```

Los argumentos usados serán:

install	Indica que el proceso que se va a realizar es la instalación de un paquete.
-y	Indica que conteste automáticamente si a las preguntas que pudieran salir en el proceso de instalación y de esta manera realizar la instalación de forma no interactiva.
openssh-server	Es el nombre del paquete a instalar.

2. Habilitar el servicio para que se inicie automáticamente al iniciar el sistema para ello se hará uso de la herramienta systemctl.

```
systemctl enable sshd
```

Los argumentos usados serán:

enable	Establece que el servicio que inicie con el arranque del sistema.
sshd	Indica el servicio que se iniciará automáticamente.

3. Iniciar el servicio mediante el uso de la herramienta systemctl.

```
systemctl start sshd
```

Los argumentos usados serán:

start	Indica que se incide un servicio.
sshd	Indica el servicio que se iniciará.

Tras estos pasos el fichero de configuración se encuentra en la siguiente ruta:

```
/etc/ssh/sshd_config
```

Servidor VNC en Windows

Como servidor VNC en sistema Windows se hará uso del servidor ThingVNC.

Para realizar la configuración se ha de seguir los siguientes pasos como administrador:

1. Descargar el instalador de la página https://www.tightvnc.com/download.php.
2. Realizar la instalación marcando las siguientes opciones.

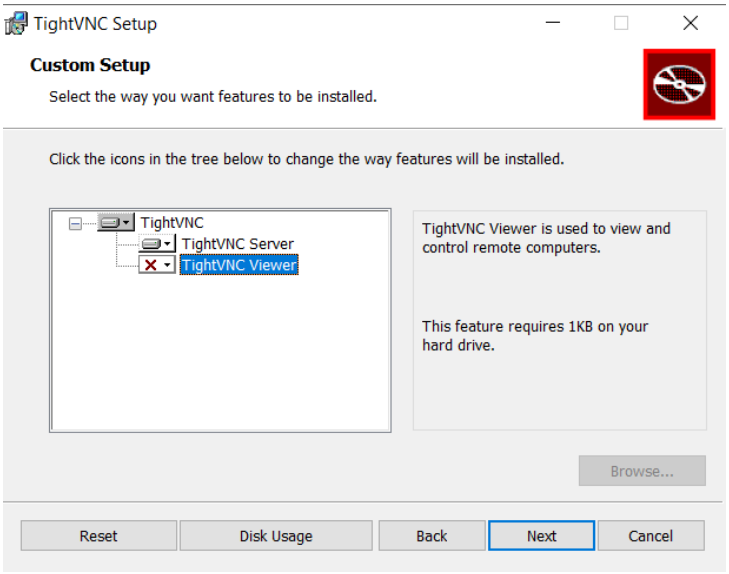

Ilustración J-0-1Selección de la instalación del servidor ThingVNC.

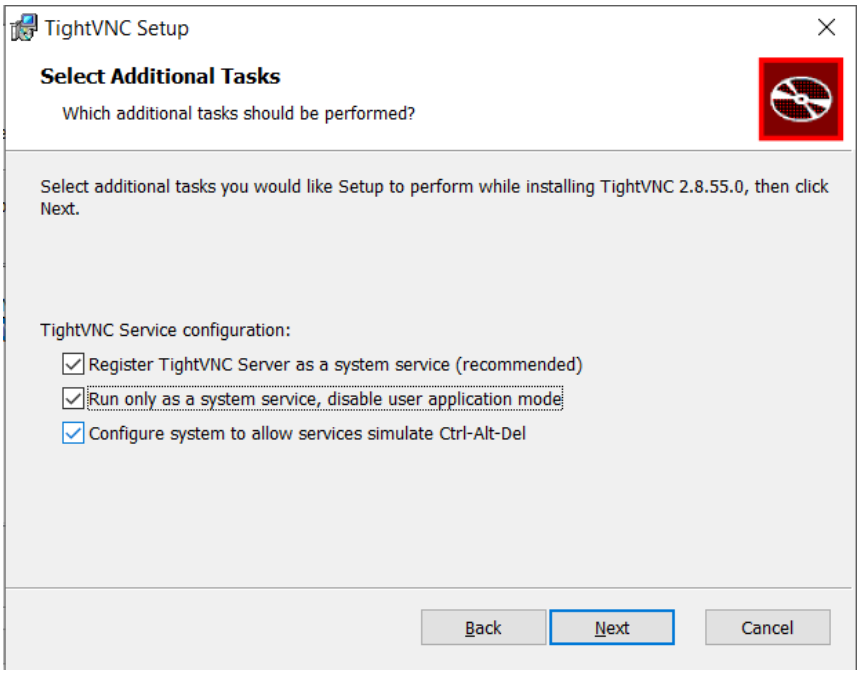

Ilustración J-0-2 Selección de ThingVNC como servicio que se autoinicie

Durante la instalación se solicitará introducir las contraseñas, solo se debe habilitar la contraseña de control de acceso.

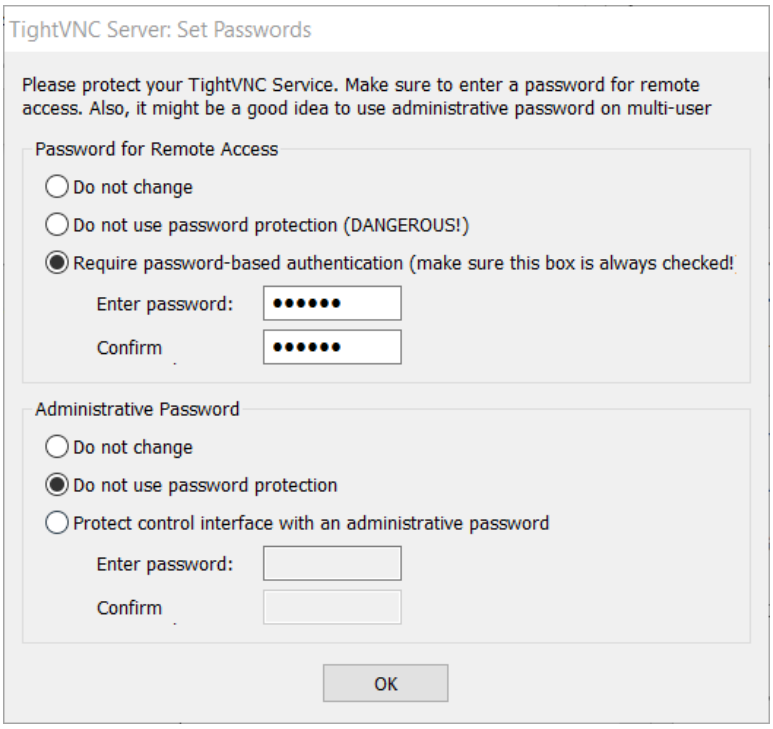

Ilustración J-0-3 Configuraccion de las contraseñas del servidor ThingVNC

3. Realizar la configuración tras la instalación, para ello se debe buscar el acceso directo TightVNC Service - Offline Configuration y se debe replicar la siguiente configuración

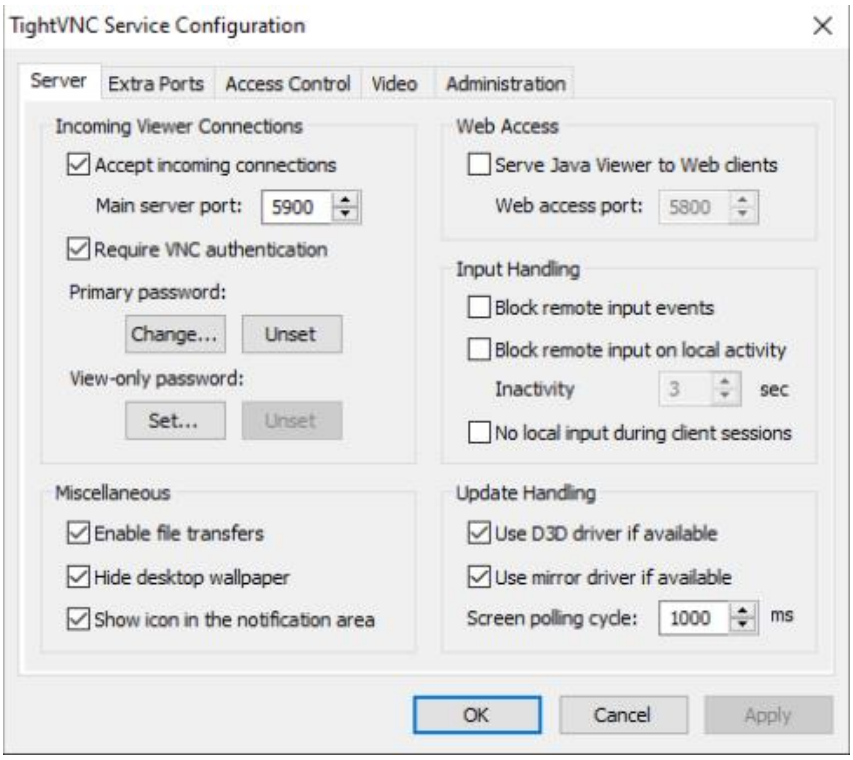

Ilustración J-0-4 Parámetros de configuración de ThingVNC pestaña de servidor.

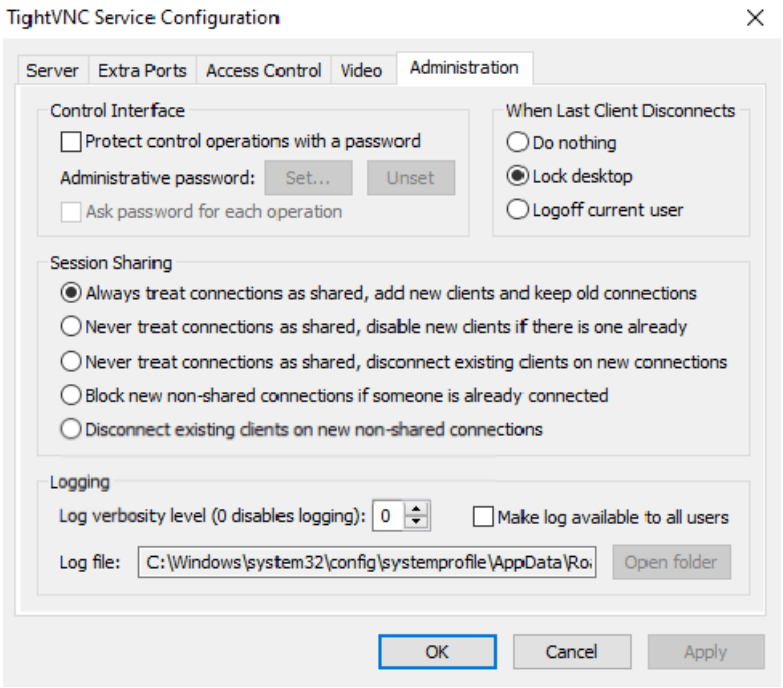

Ilustración J-0-5 Parámetros de configuración de ThingVNC pestaña de administración

Servidor RDP en Windows

Se habilitará el control remoto nativo de Windows y se hará uso de la herramienta rdpwrap, que se trata de un proyecto de GitHub ubicado en el repositorio stascorp/rdpwrap que permite la conexión multiusuario.

Esta conexión no compartirá el escritorio, aunque se inicie se sesión con el mismo usuario.

Para realizar la configuración se ha de seguir los siguientes pasos como administrador:

1. Acceder al símbolo de sistema como administrador y ejecutar el siguiente script

```
 sc config RemoteRegistry start=auto
 net start remoteregistry
 reg                                                              add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server"
/v fDenyTSConnections /t REG_DWORD /d 0 /f
```

Con este script se está replicando la siguiente configuración realizada gráficamente

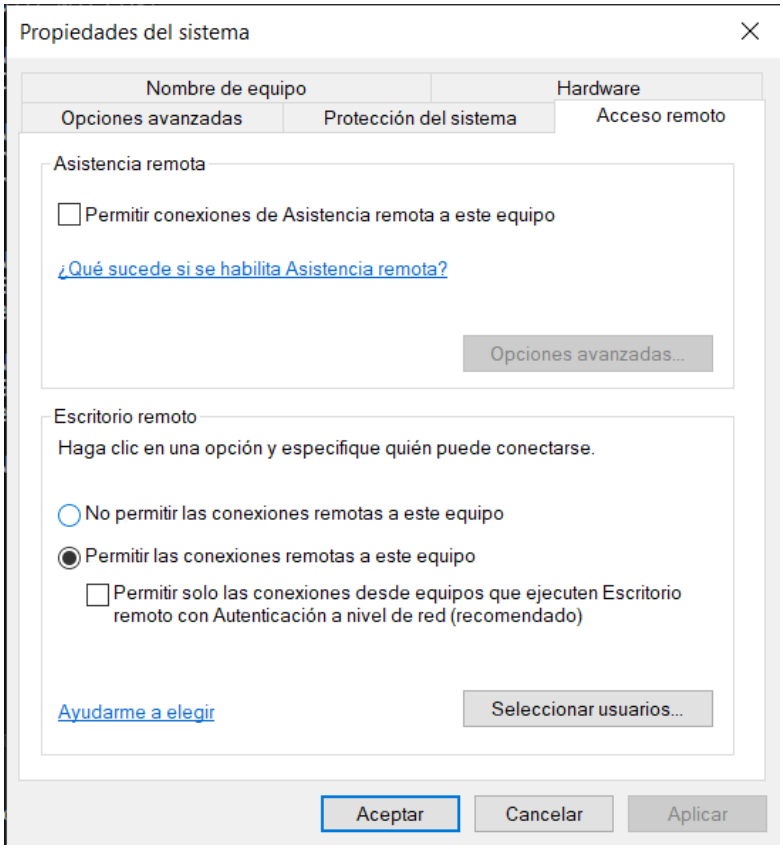

Ilustración J-0-6 Configuración grafica de RDP en Windows

2. Descargar el archivo comprimido de RDPWrap de la URL https://github.com/stascorp/rdpwrap/releases/download/v1.6.2/RDPWrap-v1.6.2.zip
3. Descomprimir el fichero RDPWrap-v1.6.2.zip.
4. Ejecutar el script install.bat como administrador.
5. Realizar la siguiente configuración lanzado RDPConf.exe.

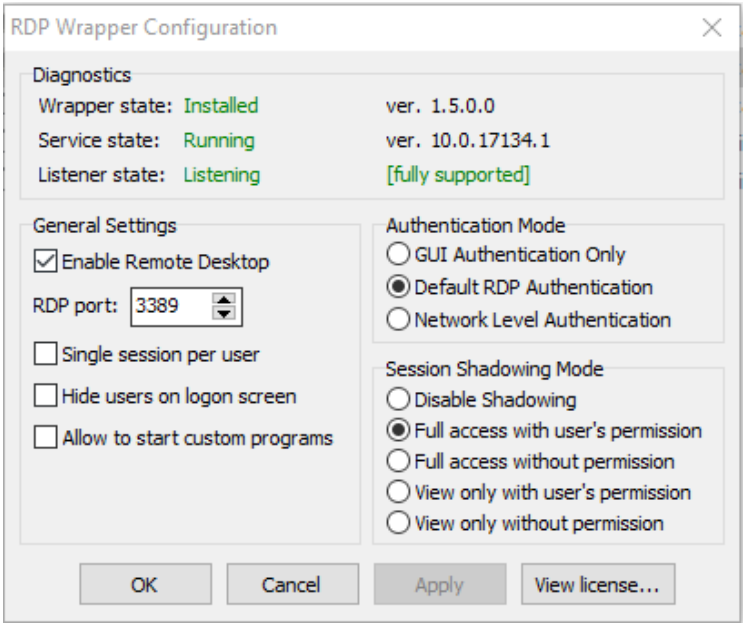

Ilustración J-0-7 Configuración grafica de RDPWrapper en Windows

Servidor SSH en Windows

El proceso se detallará únicamente para Windows 10 ya que es usado actualmente en el laboratorio de Telemática y debido a que el proceso entre diferentes versiones de Windows es totalmente distinto.

Para realizar la configuración se ha de seguir los siguientes en una terminar de PowerShell como administrador:

1. Instalar el servidor de OpenSSH, para ello es necesario añadir una nueva característica de Windows mediante el uso de la herramienta Add-WindowsCapability[124]:

```
Add-WindowsCapability -Online -Name OpenSSH.Server~~~~0.0.1.0
```

Los argumentos usados son:

-Online	Indica que se va a añadir al equipo actual.
-Name OpenSSH.Server~~~~0.0.1.0	Indica el nombre de la característica a añadir

2. Iniciar el servidor SSH, para ello uso de la herramienta Start-Service[125]:

```
Start-Service sshd
```

El argumento usado es:

sshd	Nombre del servicio a activar.

3. Configurar el autoinicio del servidor SSH, para ello uso de la herramienta Set-Service[126]:

```
Set-Service -Name sshd -StartupType 'Automatic'
```

Los argumentos usados son:

-Name sshd	Indica el nombre del servicio a configurar-
-StartupType 'Automatic'	Indica que como modalidad de inicio se seleccione la modalidad automática.

Tras estos pasos el fichero de configuración se encuentra en la siguiente ruta:

```
C:\Windows\System32\OpenSSH\sshd_config_default
```

ANEXO K: ACCESO MEDIANTE VPN

El anexo se detallará el contenido de los ficheros de configuración de los servidores y clientes OpenVPN para la solución elegida y el contenido y configuración del servicio para la creación de los bridges necesario.

Fichero de configuración servidor VPN

A continuación, se detallará el contenido de los ficheros de configuración /etc/openvpn/server_ltA.conf y /etc/openvpn/server_ltB.conf. Para ello explicar se tomará como ejemplo el fichero /etc/openvpn/server_ltA.conf.

El contenido del fichero /etc/openvpn/server_ltA.conf será:

```
  dev-type tap
  dev tapa0
  mode server
  proto tcp
  port 22
  comp-lzo
  keepalive 10 120
  server-bridge
  push "route 172.16.17.128 255.255.255.128 172.16.17.125"
  tls-server
  dh dh4096.pem
  duplicate-cn
  client-to-client
  client-config-dir /etc/openvpn/client
  ccd-exclusive
  crl-verify /etc/openvpn/easy-rsa/keys/crl.pem
  status /var/log/openvpn-status_ltA.log
  log-append /var/log/openvpn_ltB.log
  auth SHA512
  cipher AES-256-CBC
  tls-version-min 1.2
  tls-cipher     TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-
AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-256-CBC-SHA:TLS-DHE-RSA-WITH-
CAMELLIA-256-CBC-SHA:TLS-DHE-RSA-WITH-AES-128-CBC-SHA:TLS-DHE-RSA-
WITH-CAMELLIA-128-CBC-SHA
  persist-key
  persist-tun
  user nobody
  group nogroup
  ca /etc/openvpn/easy-rsa/keys/ca_ltA.crt
  cert /etc/openvpn/easy-rsa/keys/server_ltA.crt
  key /etc/openvpn/easy-rsa/keys/server_ltA.key
```

Los parámetros de configuración usados son:

dev-type tap	Indica que se haga uso de interfaz de tipo TAP para la conexión de los clientes en el servidor.
dev tapa0	Indica el nombre de la interfaz TAP donde se conectará el cliente.

mode server	Indica que la configuración se pertenece a un servidor.
proto tcp	Indica el protocolo usado para la comunicación.
port 22	Indica el número de puerto donde se expondrá el servidor VPN.
comp-lzo	Indica que se hará uso de compresión para los paquetes.
keepalive 10 120	Indica cada cuanto tiempo se comprobará el estado de cliente y mantener activa la conexión. En este caso cada 10 segundos mandará un ping y si no llegará respuesta lo intentará después de 120.
server-bridge	Indica que el servidor será usado en modo bridge. Este hará de puente entre el cliente y la subred.
push "route 172.16.17.128 255.255.255.128 172.16.17.125"	Esta configuración hace que se envíe al iniciar la configuración una entrada a la tabla de enrutamiento. En este caso se envía la ruta hasta la dirección IP de la interfaz de la subred para que petición DHCP se envíe correctamente por la interfaz de OpenVPN del cliente.
tls-server	Indica el uso TLS y que el servidor asumirá el rol de servidor durante el protocolo de enlace TLS.
dh dh4096.pem	Indica el fichero que contiene las claves Diffie Hellman.
duplicate-cn	Indica que mismo cliente puede conectarse múltiples veces de forma simultánea.
client-to-client	Indica que los clientes se puedan comunicar entre ellos.
client-config-dir /etc/openvpn/client	Indica la ubicación de la configuración especifica de cada cliente
ccd-exclusive	Indica que es obligatoria la existencia del fichero de configuración para el cliente.
crl-verify /etc/openvpn/easy-rsa/keys/crl.pem	Indica el fichero donde se encuentra los certificados revocados.
status /var/log/openvpn-status_ltA.log	Indica el fichero de log donde se almacenará el estado del servidor.
log-append /var/log/openvpn_ltB.	Indica el fichero de log los eventos ocurrido en el servidor.
auth SHA512	Indica el algoritmo usado para realizar la autentificación de los paquetes.
cipher AES-256-CBC	Indica el algoritmo usado para realizar el cifrado de los paquetes.
tls-version-min 1.2	Indica la versión mínima permitida para el protocolo TLS.
tls-cipher …	Indica una lista con los cifrados permitido para el protocolo TLS.
persist-key	Indica que solo se lea las claves al inicio del servidor, esto es necesario si se

	hace uso de los parámetros user y group. De no hacer uso seguramente no se tenga los permisos necesarios para la lectura.
persist-tun	Indica que se mantenga abierto la interfaz TAP o TUN incluso después de la desconexión de todos los clientes, esto es necesario si se hace uso de los parámetros user y group. De no hacer uso seguramente no se tenga los permisos necesarios para la lectura.
user nobody	Indica el usuario con el que se ejecutará el proceso de OpenVPN.
group nogroup	Indica el grupo con el que se ejecutará el proceso de OpenVPN.
ca /etc/openvpn/easy-rsa/keys/ca_ltA.crt	Indica la ubicación del fichero de autentificación.
cert /etc/openvpn/easy-rsa/keys/server_ltA.crt	Indica la ubicación del certificado del servidor.
key /etc/openvpn/easy-rsa/keys/server_ltA.key	Indica la ubicación de los claves del servidor.

Los cambios entre la configuración entre /etc/openvpn/server_ltA.conf y /etc/openvpn/server_ltB.conf son los siguientes:

- Se sustituye la cadena ltA por ltB.
- Se cambia el interfaz tapa0 por tapb0 al ser la interfaz TAP que se encuentra en la subred B.
- Se cambia 172.16.17.125 por 172.16.17.253 al ser esta la IP asignada a la interfaz eth2.

Fichero de configuración cliente básico VPN

A continuación, se detallará el contenido de los fiche ros de configuración base /etc/openvpn/client/cliente_ltA.conf y /etc/openvpn/client/cliente_ltB.conf. Para ello explicar se tomará como ejemplo el fichero /etc/openvpn/client/cliente_ltA.conf.

El contenido del fichero /etc/openvpn/client/cliente_ltA.conf será:

```
client
dev tapa0
proto tcp
dev-type tap
remote ait08.us.es 22
tls-client
keepalive 10 120
comp-lzo
pull
nobind
user nobody
group nogroup
persist-key
persist-tun
auth SHA512
cipher AES-256-CBC
```

```
  tls-version-min 1.2
  tls-cipher     TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-
AES-128-GCM-SHA256:TLS-DHE-RSA-WITH-AES-256-CBC-SHA:TLS-DHE-RSA-WITH-
CAMELLIA-256-CBC-SHA:TLS-DHE-RSA-WITH-AES-128-CBC-SHA:TLS-DHE-RSA-
WITH-CAMELLIA-128-CBC-SHA
  mssfix
  remote-cert-tls server
  log-append /var/log/openvpn.log
  status /var/log/openvpn-status.log
```

Los parámetros de configuración usados son:

client	Indica que la configuración se pertenece a un cliente.
dev tapa0	Indica el nombre de la interfaz TAP donde contra la que se conectará en el servidor.
proto tcp	Indica el protocolo usado para la comunicación.
dev-type tap	Indica que se haga uso de interfaz de tipo TAP para el servidor.
remote ait08.us.es 22	Inidca la ubucacion y el puerto del servidor OpenVPN.
tls-client	Indica el uso TLS y que el cliente asumirá el rol de cliente durante el protocolo de enlace TLS.
keepalive 10 120	Indica cada cuanto tiempo se comprobará el estado de cliente y mantener activa la conexión. En este caso cada 10 segundos mandará un ping y si no llegará respuesta lo intentará después de 120.
pull	Indica que hará uso de la configuración proporcionada por el servidor.
nobind	Indica que no se vincule el proceso a una dirección local. Necesario debido a que la dirección IP será asignada mediante DHCP.
user nobody	Indica el usuario con el que se ejecutará el proceso de OpenVPN. Se ignorará en cliente Windows.
group nogroup	Indica el grupo con el que se ejecutará el proceso de OpenVPN. Se ignorará en cliente Windows.
persist-key	Indica que solo se lea las claves al inicio del servidor, esto es necesario si se hace uso de los parámetros user y group. De no hacer uso seguramente no se tenga los permisos necesarios para la lectura.
persist-tun	Indica que se mantenga abierto la interfaz TAP o TUN incluso después de la desconexión del servidor para el intento de reconexión, esto es necesario si se hace uso de los parámetros user y group. De no hacer uso seguramente no se tenga los permisos necesarios para la lectura.
auth SHA512	Indica el algoritmo usado para realizar la autentificación de los paquetes.
cipher AES-256-CBC	Indica el algoritmo usado para realizar el cifrado de los paquetes.
tls-version-min 1.2	Indica una lista con los cifrados permitido para el protocolo TLS.

tls-cipher …	Indica la versión mínima permitida para el protocolo TLS.
mssfix	Indica que se inicie un proceso al establecer la conexión para evitar fallo por MTU.
remote-cert-tls server	Indica que el certificado será verificado en el servidor.
log-append /var/log/openvpn.log	Indica el fichero de log los eventos ocurrido en el cliente. Se debe comentar para cliente Windows.
status /var/log/openvpn-status.log	Indica el fichero de log donde se almacenará el estado del cliente. Se debe comentar para cliente Windows.

Solo se deberá cambiar la interfaz tapa0 por tapb0 para generar el fichero /etc/openvpn/client/cliente_ltB.conf.

Servicio para la creación de dispositivos Bridge para OpenVPN

Se va a detallará todos archivos necesarios para la configuración de los bridges necesario para la solución elegida para el acceso VPN.

Se deberá tener la siguiente estructura de fichero:

- /opt/bridgeScript
    - detenerBridges.sh
    - levantarBridges.sh
        - configuracionBridge.cfg
        - destruirBridge.sh
        - levantarBridge.sh
    - ltB
        - configuracionBridge.cfg
        - destruirBridge.sh
        - levantarBridge.sh
- /etc/systemd/system/bridgeOpenvpn.service

Todos esto ficheros serán entregado junto con la memoria.

A continuación, se detallará el funcionamiento de cada uno de los scripts y el contenido de los ficheros de configuración.

Script detenerBridges.sh

Se trata de un script sencillo se encargará únicamente de llamar a los scripts /opt/bridgeScript/ltA/destruirBridge.sh y /opt/bridgeScript/ltB/destruirBridge.sh y comprobar el resultado devuelto por los mismo.

Script levantarBridges.sh

Se trata de un script sencillo se encargará únicamente de llamar a los scripts /opt/bridgeScript/ltA/levantarBridge.sh y /opt/bridgeScript/ltB/levantarBridge.sh y comprobar el resultado

devuelto por los mismo.

Script destruirBridge.sh

Se procederá a explicar el funcionamiento y pasos que se realiza.

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobar que este instalada la dependencia bridge-utils.

A continuación, se comprueba que exista el fichero de configuración configuracionBridge.cfg en el mismo directorio donde se encuentra el y este contenga las variables interfaz_red, interfaz_bridge, interfaces_tap.

De existir se comprobará que existan las interfaces de red interfaz_red y interfaz_bridge.

Luego se procederá a parar y eliminar el bridge interfaz_bridge.

Tras esto se desasociaras todas las interfaces TAP definidas en interfaces_tap y que existieran.

Para finalizar se deshabilitará el modo promiscuo de interfaz_red y se reiniciará la interfaz.

Todas las notificaciones de errores o del progreso del proceso se realizará mediante el uso de la herramienta logger y se puede ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

Script levantarBridge.sh

Se procederá a explicar el funcionamiento y pasos que se realiza.

Para comenzar se comprobará que se está ejecutando como privilegios de superusuario(root).

Después el script comprobar que este instalada la dependencia bridge-utils.

A continuación, se comprueba que exista el fichero de configuración configuracionBridge.cfg y este contenga las variables interfaz_red, interfaz_bridge, interfaces_tap y is_gateway (se verificará que su valor sea TRUE o FALSE).

De existir se comprobará que existan la interfaz de red interfaz_red.

Luego se verificará que esta interfaz solo disponga una dirección IP asociada.

Posteriormente se creará los dispositivos TAP definidos en interfaces_tap y se asociará a OpenVPN.

Tras esto creará el bridge interfaz_bridge y se asociará a es tanto la interfaz interfaz_red como las interfaces interfaces_tap. En este proceso se habilitará el modo promiscuo para las interfaces.

Para finalizar se eliminará la dirección IP de la interfaz interfaz_red y se asociará al bridge interfaz_bridge.

En caso de que el valor de la variable is_gateway fuera TRUE antes de terminar la ejecución del script se cambiará la ruta por defecto para que este asociado al bridge interfaz_bridge.

Todas las notificaciones de errores o del progreso del proceso se realizará mediante el uso de la herramienta logger y se puede ver en el archivo log de syslog del sistema y en la salida estándar. Los logs se identificarán por el nombre del script.

Script configuracionBridge.cfg

El contenido del fichero /opt/bridgeScript/ltA/configuracionBridge.cfg es:

```
interfaz_red="eth1"
interfaz_bridge="breth1"
```

```
interfaces_tap="tapa0"
is_gateway="FALSE"
```

El contenido del fichero /opt/bridgeScript/ltA/configuracionBridge.cfg es:

```
interfaz_red="eth2"
interfaz_bridge="breth1"
interfaces_tap="tapb0"
is_gateway="FALSE"
```

Las variables definidas en estos ficheros son:

interfaz_red	Nombre del interfaz de red que formara parte del bridge que se va a crear.
interfaz_bridge	Nombre del bridge que se va a crear.
interfaces_tap	Define la lista de dispositivos TAP que se incluirá en el bridge. Ejemplo de la definición de más de un elemento interface_tap="tap0 tap1 tap2".
is_gateway	Indica si interfaz_red es la interfaz conectada a la ruta predeterminada. Los posibles valores son TRUE o FALSE.

Servicio bridgeOpenvpn.service

Este define un servicio que será usado para que al inicio del sistema se creen el bridge necesario para la configuración elegida para OpenVPN y que esto se realice antes del inicio de los servicios de los servidores OpenVPN.

El contenido del fichero del servicio es:

```
[Unit]
Description=Bridge para openvpn
Before=openvpn.service
Before=openvpn@server_ltA.service
Before=openvpn@server_ltB.service


[Service]
Type=oneshot
ExecStart=/opt/bridgeScript/levantarBridges.sh
RemainAfterExit=true
ExecStop=/opt/bridgeScript/detenerBridges.sh
StandardOutput=journal

[Install]
WantedBy=multi-user.target
```

Los campos se usado para el servicio son:

Description=Bridge para openvpn	Establece una descripción corta del servicio.
Before=openvpn.service	Indica que servicio debe iniciarse antes de poder iniciar este servicio.

Before=openvpn@server_ltA.service Before=openvpn@server_ltB.service	
Type=oneshot	Establece que por el tipo de servicio el gestor de servicio interpretará como iniciado el servicio una vez haya finalizado el proceso lanzado al iniciarlo. Por este mismo motivo es necesario la variable RemainAfterExit para indicar que debe ejecutar el comando de para antes de volver a iniciar el servicio.
ExecStart=comando	Indica el comanda a ejecutar cuando se inicia el servicio.
ExecStop=comando	Indica el comanda a ejecutar cuando se para el servicio.
RemainAfterExit=true	Establece que el servicio debe pararse ante de volver a intentar arrancarlo.
WantedBy=multi-user.target	Indica que no se inicie el servicio si el sistema no se encuentra en el nivel de ejecución igual o superior a donde este habilitado el servicio de red.

ANEXO L: SCRIPTS DE GESTIÓN DE CONMUTADORES

El anexo se detallará los procesos necesarios para realizar la configuración para activar el servicio ssh, creación de usuario administrador y borrado de configuración total.

Activar servicio ssh

Conmutador Cisco 3750G-24TS

Para activar ssh se debe configurar una dirección IP que sea accesible que nos permita hacer funciones de gestión (no tiene porque que estar fuera del enrutamiento IP). Para ello se debe ejecutar los siguientes comandos como ejemplo:

```
enable
configure t
interface vlan 1
ip address 192.168.0.2 255.255.255.0
exit
```

Luego de debe configurar un nombre y un nombre de dominio para ello se puede seguir el siguiente ejemplo:

```
enable
configure t
hostname SWITCH_CISCO
ip domain name SWITCH_CISCO
```

Posteriormente se debe generar las claves RSA necesaria para la conexión ssh y para ello hay que ejecutar el siguiente comando y cuando se solicite el número de bits de la clave hay que especificar como mínimo 1024:

```
enable
configure t
crypto key generate rsa
#Al preguntar How many bits in the modulus [512]:
1024
```

Para continuar se debe activar el servicio ssh y la transferencia de archivos mediante ssh. Para ello se debe ejecutar los siguientes comandos:

```
enable
configure t
ip ssh version 2
ip scp server enable
```

El último paso que debe seguir será crear una línea virtual para permitir la conexión ssh. Los comandos que se

debe seguir son:

```
enable
configure
line vty X Y
transport input ssh
login local
exit
```

Siendo X un número de 0 – 4 que indica que línea se usara para la conexión ssh e Y un numero de 0 - 15 que indica el número de conexiones simultanea siendo 1 si indicamos 0 y 16 si indicamos 15.

El firmware por defecto no incluye la posibilidad de activar ssh, por ello es necesario como mínimo 12.2(58)SE1[127].

Conmutador HP 2620-24 o 2650-24

Para activar ssh es necesario configurar un IP que sea accesible que nos permita hacer funciones de gestión (no tiene porque que estar fuera del enrutamiento IP). Para ello se debe ejecutar los siguientes comandos como ejemplo:

```
enable
configure
vlan 1
ip address 192.168.0.2 255.255.255.0
exit
```

Para continuar se debe activar el servicio ssh y la transferencia de archivos mediante ssh. Para ello se debe ejecutar los siguientes comandos como ejemplo:

```
enable
configure
ip ssh
ip ssh filetransfer
```

Al habilitar la transferencia de ficheros mediante ssh se deshabilita el demonio de tftp y el cliente, para habilitarlo de nuevo se puede ejecutar lo siguiente comandos, pero esto deshabilitará la transferencia de ficheros mediante ssh:

```
enable
configure
tftp client
tftp server
```

Debido a la implementación de ssh que disponemos en estos conmutadores solo disponemos del cifrado 3des-cbc compatible entre las dos modelos de conmutador.

Creación de usuario administrador

Conmutador Cisco 3750G-24TS

Para crear un usuario con privilegio de administrador se deben ejecutar los siguientes comandos:

```
enable
configure
username USERNAME privilege 15 secret PASSWORD
enable secret PASSWORD
exit
```

Para únicamente poner contraseña para habilitar los privilegios mediante el comando:

```
enable
```

Se deben ejecutar los siguientes comandos:

```
enable
configure t
enable secret PASSWORD
```

Conmutador HP 2620-25 o 2650-25

Para crear un usuario con privilegio de administrador se deben ejecutar los siguientes comandos

```
enable
configure
password manager user-name USERNAME
#Al preguntar New password for manager:
PASSWORD
#Al preguntar Re-enter the new password for manager:
PASSWORD
exit
```

Borrado de la configuración total

Conmutador Cisco 3750G-24TS

Los pasos se deben seguir son:

1. Desconectar con el conmutador de la corriente.

1. Pulsar el botón mode y conectar el cable de corriente.
2. Mantener unos 15 segundo hasta que el led de SYST parpadee en naranja y luego vuelva a verde y se mantenga en sin parpadear
3. Esperar a que inicie el conmutador y muestre una consola como esta

```
Boot Sector Filesystem (bs) installed, fsid: 2
Base ethernet MAC Address: 30:37:a6:01:f4:80
```

```
Xmodem file system is available.
The password-recovery mechanism is enabled.

The system has been interrupted prior to initializing the
flash filesystem.  The following commands will initialize
the flash filesystem, and finish loading the operating
system software:

    flash_init
    boot


switch:
```

4. Introducir el comando flash_init
5. Ejecutar el comando set que mostrará las variables configurada para el arranque de conmutador

```
BAUD=115200
BOOT=flash:/c3750_15.bin
BOOTHLPR=flash:/c3750_15.bin
BOOT_MANUAL=
DEFAULT_ROUTER=
HELPER=
IP_ADDR=
MANUAL_BOOT=no
PRIV_CONFIG=flash:/private-config.text
SDM_TEMPLATE_ID=0
SWITCH_NUMBER=1
SWITCH_PRIORITY=1
```

De aquí se puede ver que en el archivo flash:/private-config.text estaría la configuración privada.

6. Borrar o renombrarlos siguientes ficheros terminado en flash:*text y el archivo flash:config.text con el comando delete o rename. Para ver que fichero es necesario borrar o renombrar podemos ejecutar

```
dir flash:
```

Para cada archivo nos pedirá una confirmación al ejecutar el comando ejecutado.

También se debe que borrar las variables que apuntaba a un archivo, para ver las variables se debe ejecutar:

```
set
```

Un ejemplo de borrado sería:

```
set  PRIV_CONFIG
```

7. Ejecutar el comando boot.

- Si está declarada anteriormente BOOTHLPR y no BOOT reiniciaremos el switch sin configuración.
- Si no estaba declarada la variable BOOTHLPR y BOOT cogerá la imagen por defecto y si no hubiere deberemos setear nosotros la ruta hasta la imagen de firmware

```
set BOOTHLPR flash:/c3750_15.bin
```

Nota los bauds para la conexión depende de si se configuraron anteriormente en el conmutador por defecto son 9600 pero una vez configurado esto también afecta al modo de recuperación.

Conmutador HP 2620-25 o 2650-25

Los pasos se deben seguir son:

1. Mantener presionado el botón Reset.
2. Mientras mantiene presionado el botón Reset, presione y mantener presionado el botón Clear
3. Soltar el botón Reset.
4. Cuando el LED de TEST a la derecha del Clear comienza a parpadear, sueltar el botón Clear.

El conmutador tarda aproximadamente 20-25 segundos en reiniciarse.

Si esta deshabilitado la configuración front-panel-security factory-reset no se podrá usar la combinación de botones para hacer el borrado completo de conmutador.

Aun así, hay una posibilidad de recuperar la contraseña si la versión de ROM del conmutador permite al arrancar el conmutador y estando conectado por el puerto serie a 9600 baud nos sale el siguiente menú:

```
Boot Profiles:

0. Monitor ROM Console
1. Primary Software Image
2. Secondary Software Image

Select profile (primary): 0
```

Entonces se debe acceder a la opción Monitor ROM Console y ejecutar el siguiente comando:

```
cat  cfa0/mgrinfo.txt
```

Y mostrará un resultado como una de estas opciones:

- Si tenemos contraseña tanto para operator como para manager.

  El nombre de usuario manager seria -MANAGER- y la contraseña _PASSMANAGER_

  El nombre de usuario operator seria -OPERATOR- y la contraseña _PASSOPERATOR_

```
FLG_PASSMANAGER_FLG_PASSOPERATOR_▒▒▒▒▒-MANAGER--OPERATOR-
_PASSMANAGER__PASSMANAGER_-MANAGER—OPERATOR-▒
```

- Si tenemos contraseña solo para operator.

  El nombre de usuario operator seria -OPERATOR- y la contraseña _PASSOPERATOR_

```
FLG_PASSOPERATOR_▒▒▒▒▒-OPERATOR-_ PASSOPERATOR _- -OPERATOR -▒
```

- Si tenemos contraseña solo para manager.

El nombre de usuario manager seria -MANAGER- y la contraseña _PASSMANAGER_

```
FLG_PASSMANAGER_ ▒▒▒▒▒-MANAGER-_PASSMANAGER_ -MANAGER-▒
```

Para más información https://techhub.hpe.com/eginfolib/networking/docs/switches/WB/15-18/5998-8152_wb_2920_asg/content/ch02s05.html

REFERENCIAS

[1] "Characteristics of Linux RAID levels." https://linux.die.net/EVMSUG/characraidlvls.html (accessed Jul. 19, 2020).

[2] "HP Dynamic Smart Array B140i Controller - Overview." https://support.hpe.com/hpesc/public/docDisplay?docLocale=en_US&docId=c04406959 (accessed Jul. 19, 2020).

[3] "mdadm(8) - Linux manual page." https://www.man7.org/linux/man-pages/man8/mdadm.8.html (accessed Jul. 25, 2020).

[4] "Ubuntu Manpage: fsarchiver - filesystem archiver." http://manpages.ubuntu.com/manpages/xenial/man8/fsarchiver.8.html (accessed Jul. 25, 2020).

[5] "dd(1) - Linux manual page." https://man7.org/linux/man-pages/man1/dd.1.html (accessed Jul. 25, 2020).

[6] "sfdisk(8) - Linux manual page." https://man7.org/linux/man-pages/man8/sfdisk.8.html (accessed Jul. 25, 2020).

[7] "blkid(8) - Linux manual page." https://man7.org/linux/man-pages/man8/blkid.8.html (accessed Jul. 25, 2020).

[8] "mkswap(8) - Linux manual page." https://man7.org/linux/man-pages/man8/mkswap.8.html (accessed Jul. 26, 2020).

[9] "grub-install(8): install GRUB on your drive - Linux man page." https://linux.die.net/man/8/grub-install (accessed Jul. 26, 2020).

[10] "mount(8) - Linux manual page." https://www.man7.org/linux/man-pages/man8/mount.8.html (accessed Jul. 26, 2020).

[11] "mkdir(1) - Linux manual page." https://man7.org/linux/man-pages/man1/mkdir.1.html (accessed Aug. 01, 2020).

[12] "LVM (Español) - ArchWiki." https://wiki.archlinux.org/index.php/LVM_(Espa%C3%B1ol)#Instalar_Arch_Linux_sobre_LVM (accessed Aug. 15, 2020).

[13] "vgcfgrestore(8) - Linux manual page." https://man7.org/linux/man-pages/man8/vgcfgrestore.8.html (accessed Aug. 15, 2020).

[14] "vgcfgbackup(8) - Linux manual page." https://man7.org/linux/man-pages/man8/vgcfgbackup.8.html (accessed Aug. 15, 2020).

[15] "vgchange(8) - Linux manual page." https://man7.org/linux/man-pages/man8/vgchange.8.html (accessed Aug. 15, 2020).

[16] "pvcreate(8) - Linux manual page." https://man7.org/linux/man-pages/man8/pvcreate.8.html (accessed Aug. 15, 2020).

[17] "chroot(1) - Linux manual page." https://man7.org/linux/man-pages/man1/chroot.1.html (accessed Aug. 19, 2020).

[18] "Git - Reference." https://git-scm.com/docs (accessed Jul. 28, 2020).

[19] "rsync(1) - Linux manual page." https://man7.org/linux/man-pages/man1/rsync.1.html (accessed Jul. 29, 2020).

[20] "ssh(1): OpenSSH SSH client - Linux man page." https://linux.die.net/man/1/ssh (accessed Aug. 01, 2020).

[21] "ssh-keygen(1) - Linux man page." https://linux.die.net/man/1/ssh-keygen (accessed Aug. 01, 2020).

[22] "ssh-copy-id(1) - Linux man page." https://linux.die.net/man/1/ssh-copy-id (accessed Aug. 01, 2020).

[23] "cron(8) - Linux manual page." https://man7.org/linux/man-pages/man8/cron.8.html (accessed Aug. 05, 2020).

[24] "id(1) - Linux manual page." https://man7.org/linux/man-pages/man1/id.1.html (accessed Aug. 01, 2020).

[25] "cd(1p) - Linux manual page." https://man7.org/linux/man-pages/man1/cd.1p.html (accessed Aug. 02, 2020).

[26] "git-store-meta-GitHub." https://github.com/danny0838/git-store-meta (accessed Aug. 04, 2020).

[27] "chown(2) - Linux manual page." https://man7.org/linux/man-pages/man2/chown.2.html (accessed Aug. 02, 2020).

[28] "rm(1) - Linux manual page." https://man7.org/linux/man-pages/man1/rm.1.html (accessed Aug. 02, 2020).

[29] "date(1) - Linux manual page." https://man7.org/linux/man-pages/man1/date.1.html (accessed Aug. 03, 2020).

[30] "less(1) - Linux manual page." https://man7.org/linux/man-pages/man1/less.1.html (accessed Aug. 05, 2020).

[31] "scp(1) - Linux manual page." https://man7.org/linux/man-pages/man1/scp.1.html (accessed Aug. 11, 2020).

[32] "chattr(1) - Linux manual page." https://man7.org/linux/man-pages/man1/chattr.1.html (accessed Aug. 11, 2020).

[33] "tar(1) - Linux manual page." https://man7.org/linux/man-pages/man1/tar.1.html (accessed Aug. 11, 2020).

[34] "md5sum(1) - Linux manual page." https://man7.org/linux/man-pages/man1/md5sum.1.html (accessed Aug. 11, 2020).

[35] "diff(1) - Linux manual page." https://man7.org/linux/man-pages/man1/diff.1.html (accessed Aug. 12, 2020).

[36] "ls(1) - Linux manual page." https://man7.org/linux/man-pages/man1/ls.1.html (accessed Aug. 12, 2020).

[37] "grep(1) - Linux manual page." https://man7.org/linux/man-pages/man1/grep.1.html (accessed Aug. 12, 2020).

[38] "cat(1) - Linux manual page." https://man7.org/linux/man-pages/man1/cat.1.html (accessed Aug. 12, 2020).

[39] "Guía de usuario de HPE iLO 4." https://support.hpe.com/hpesc/public/docDisplay?docId=c03334053 (accessed Aug. 25, 2020).

[40] "HPE Integrated Lights-Out (iLO)." https://support.hpe.com/hpesc/public/docDisplay?docLocale=en_US&docId=c04154343 (accessed Aug. 25, 2020).

[41] "Guía de usuario del servidor HP ProLiant ML310e Gen8." https://support.hpe.com/hpesc/public/docDisplay?docId=emr_na-c03514412 (accessed Aug. 26, 2020).

[42] "¿Qué es KVM?" https://www.redhat.com/es/topics/virtualization/what-is-KVM (accessed Aug. 27, 2020).

[43] "QEMU." https://www.qemu.org/ (accessed Aug. 28, 2020).

[44] "libvirt: The virtualization API." https://libvirt.org/index.html (accessed Aug. 29, 2020).

[45] "libvirt: virsh." https://libvirt.org/manpages/virsh.html (accessed Aug. 28, 2020).

[46] "virt-viewer(1) - Linux man page." https://linux.die.net/man/1/virt-viewer (accessed Aug. 29, 2020).

[47] "Spice User Manual." https://www.spice-space.org/spice-user-manual.html (accessed Aug. 29, 2020).

[48] "dialog(1): dialog boxes from shell scripts - Linux man page." https://linux.die.net/man/1/dialog (accessed Aug. 28, 2020).

[49] "libvirt: Domain XML format." https://libvirt.org/formatdomain.html (accessed Aug. 29, 2020).

[50] "Pacemaker - ClusterLabs." https://wiki.clusterlabs.org/wiki/Pacemaker (accessed Nov. 24, 2020).

[51] "Corosync by corosync." https://corosync.github.io/corosync/ (accessed Nov. 24, 2020).

[52] "DRBD » LINBIT." https://www.linbit.com/drbd/ (accessed Nov. 24, 2020).

[53] "pcs - pacemaker/corosync configuration system - man page." https://www.mankier.com/8/pcs (accessed Dec. 08, 2020).

[54] "Bienvenido a OpenGnsys | OpenGnsys." https://opengnsys.es/web/ (accessed Nov. 24, 2020).

[55] "iptables(8) - Linux manual page." https://man7.org/linux/man-pages/man8/iptables.8.html (accessed Nov. 24, 2020).

[56] "apt-get(8) - Linux man page." https://linux.die.net/man/8/apt-get (accessed Oct. 05, 2020).

[57] "1.3. Configuring the iptables Firewall to Allow Cluster Components Red Hat Enterprise Linux 7 | Red Hat Customer Portal." https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/high_availability_add-on_reference/s1-firewalls-haar (accessed Nov. 25, 2020).

[58] "systemctl(1) - Linux manual page." https://www.man7.org/linux/man-pages/man1/systemctl.1.html (accessed Sep. 06, 2020).

[59] "passwd(1) - Linux manual page." https://man7.org/linux/man-pages/man1/passwd.1.html (accessed Nov. 24, 2020).

[60] "ocf_heartbeat_anything." http://www.linux-ha.org/doc/man-pages/re-ra-anything.html (accessed Nov. 24, 2020).

[61] "curl(1) - Linux manual page." https://man7.org/linux/man-pages/man1/curl.1.html (accessed Nov. 24, 2020).

[62] "modprobe(8) - Linux manual page." https://man7.org/linux/man-pages/man8/modprobe.8.html (accessed Nov. 24, 2020).

[63] "drbdadm-9.0(8) — drbd-utils — Debian testing — Debian Manpages." https://manpages.debian.org/testing/drbd-utils/drbdadm-9.0.8.en.html (accessed Nov. 25, 2020).

[64] "mke2fs(8) - Linux manual page." https://man7.org/linux/man-pages/man8/mke2fs.8.html (accessed Nov. 25, 2020).

[65] "ip(8) - Linux manual page." https://man7.org/linux/man-pages/man8/ip.8.html (accessed Nov. 25, 2020).

[66] "umount(8) - Linux manual page." https://man7.org/linux/man-pages/man8/umount.8.html (accessed Nov. 25, 2020).

[67] "Ubuntu Manpage: ocf_heartbeat_IPaddr2 - Manages virtual IPv4 and IPv6 addresses (Linux specific version)." http://manpages.ubuntu.com/manpages/bionic/man7/ocf_heartbeat_IPaddr2.7.html (accessed Nov. 25, 2020).

[68] "Ubuntu Manpage: ocf_heartbeat_dhcpd - Chrooted ISC DHCP server resource agent." http://manpages.ubuntu.com/manpages/xenial/man7/ocf_heartbeat_dhcpd.7.html (accessed Nov. 25, 2020).

[69] "crontab(1) - Linux manual page." https://man7.org/linux/man-pages/man1/crontab.1.html (accessed Aug. 08, 2020).

[70] "ocf_heartbeat_apache." http://www.linux-ha.org/doc/man-pages/re-ra-apache.html (accessed Nov. 25, 2020).

[71] "ocf_linbit_drbd man page - drbd-pacemaker | ManKier." https://www.mankier.com/7/ocf_linbit_drbd (accessed Nov. 26, 2020).

[72] "ocf_heartbeat_Filesystem." http://www.linux-ha.org/doc/man-pages/re-ra-Filesystem.html (accessed Nov. 26, 2020).

[73] "systemd(1) - Linux manual page." https://www.man7.org/linux/man-pages/man1/systemd.1.html (accessed Sep. 05, 2020).

[74] "systemd.service(5) - Linux manual page." https://man7.org/linux/man-pages/man5/systemd.service.5.html (accessed Sep. 05, 2020).

[75] "xinit(1): X Window System initializer - Linux man page." https://linux.die.net/man/1/xinit (accessed Sep. 06, 2020).

[76] "i3 - improved tiling wm." https://i3wm.org/ (accessed Oct. 05, 2020).

[77] "yum(8) - Linux manual page." https://man7.org/linux/man-pages/man8/yum.8@@yum.html (accessed Oct. 05, 2020).

[78] "dirname(1) - Linux manual page." https://man7.org/linux/man-pages/man1/dirname.1.html (accessed Nov. 23, 2020).

[79] "su(1) - Linux manual page." https://man7.org/linux/man-pages/man1/su.1.html (accessed Oct. 05, 2020).

[80] "vmware Command Options." https://docs.vmware.com/en/VMware-Workstation-Pro/15.0/com.vmware.ws.using.doc/GUID-7369457F-FE1D-40FE-97B6-B29CA4916CCD.html#GUID-7369457F-FE1D-40FE-97B6-B29CA4916CCD (accessed Oct. 05, 2020).

[81] "VirtualBoxVM.exe." https://renenyffenegger.ch/notes/Windows/dirs/Program-Files/Oracle/VirtualBox/VirtualBoxVM_exe (accessed Oct. 12, 2020).

[82] "Chapter 8. VBoxManage." https://www.virtualbox.org/manual/ch08.html (accessed Oct. 12, 2020).

[83] "vncviewer(1): VNC viewer for X - Linux man page." https://linux.die.net/man/1/vncviewer (accessed Oct. 17, 2020).

[84] "nc(1): arbitrary TCP/UDP connections/listens - Linux man page." https://linux.die.net/man/1/nc (accessed Oct. 17, 2020).

[85] "timeout(1) - Linux manual page." https://man7.org/linux/man-pages/man1/timeout.1.html (accessed Oct. 17, 2020).

[86] "sleep(1) - Linux manual page." https://man7.org/linux/man-pages/man1/sleep.1.html (accessed Oct. 17, 2020).

[87] "vncpasswd(1): change VNC password - Linux man page." https://linux.die.net/man/1/vncpasswd (accessed Oct. 17, 2020).

[88] "kill(1) - Linux manual page." https://man7.org/linux/man-pages/man1/kill.1.html (accessed Oct. 18, 2020).

[89] "xfreerdp(1): FreeRDP X11 client - Linux man page." https://linux.die.net/man/1/xfreerdp (accessed Oct. 17, 2020).

[90] "udp-sender(1): broadcast file on LAN - Linux man page." https://linux.die.net/man/1/udp-sender (accessed Nov. 29, 2020).

[91] "SyncronizeConsolav1.1.0 – Proyecto OpenGnsys." https://opengnsys.es/trac/wiki/SyncronizeConsolav1.1.0 (accessed Dec. 03, 2020).

[92] "Wget(1) - Linux manual page." https://man7.org/linux/man-pages/man1/Wget.1.html (accessed Nov.

30, 2020).

[93] "mysql(1) - Linux manual page." https://man7.org/linux/man-pages/man1/mysql.1.html (accessed Nov. 30, 2020).

[94] "bash(1) - Linux manual page." https://man7.org/linux/man-pages/man1/bash.1.html (accessed Nov. 30, 2020).

[95] "DocumentacionUsuario1.0.6/Offline – Proyecto OpenGnsys." https://opengnsys.es/trac/wiki/DocumentacionUsuario1.0.6/Offline (accessed Nov. 30, 2020).

[96] "Forum List – Discussion – Proyecto OpenGnsys." https://opengnsys.es/trac/discussion/ (accessed Dec. 01, 2020).

[97] "e2fsck(8) - Linux manual page." https://man7.org/linux/man-pages/man8/e2fsck.8.html (accessed Nov. 30, 2020).

[98] "tune2fs(8) - Linux manual page." https://man7.org/linux/man-pages/man8/tune2fs.8.html (accessed Nov. 30, 2020).

[99] "resize2fs(8) - Linux manual page." https://man7.org/linux/man-pages/man8/resize2fs.8.html (accessed Nov. 30, 2020).

[100] "DocumentacionUsuario1.1.0/MenuPersonalizado – Proyecto OpenGnsys." https://opengnsys.es/trac/wiki/DocumentacionUsuario1.1.0/MenuPersonalizado (accessed Dec. 01, 2020).

[101] "Apache Guacamole™." https://guacamole.apache.org/ (accessed Dec. 02, 2020).

[102] "jcampbell1/simple-file-manager: A Simple PHP file manager. The code is a single php file." https://github.com/jcampbell1/simple-file-manager (accessed Dec. 02, 2020).

[103] "Chapter 6. Database authentication." https://guacamole.apache.org/doc/gug/jdbc-auth.html#jdbc-auth-schema (accessed Dec. 02, 2020).

[104] "Veyon – Cross-platform computer monitoring and classroom management." https://veyon.io/ (accessed Dec. 04, 2020).

[105] "La primera aplicación rápida de escritorio remoto – AnyDesk." https://anydesk.com/es (accessed Dec. 04, 2020).

[106] "Apache Tomcat® - Welcome!" http://tomcat.apache.org/ (accessed Dec. 04, 2020).

[107] "Welcome! - The Apache HTTP Server Project." https://httpd.apache.org/ (accessed Dec. 04, 2020).

[108] "Apache Guacamole™: 1.2.0." https://guacamole.apache.org/releases/1.2.0/ (accessed Dec. 04, 2020).

[109] "make(1) - Linux manual page." https://man7.org/linux/man-pages/man1/make.1.html (accessed Dec. 04, 2020).

[110] "ldconfig(8) - Linux manual page." https://man7.org/linux/man-pages/man8/ldconfig.8.html (accessed Dec. 04, 2020).

[111] "a2enmod(8) — apache2 — Debian jessie — Debian Manpages." https://manpages.debian.org/jessie/apache2/a2enmod.8.en.html (accessed Dec. 04, 2020).

[112] "a2ensite(8) — apache2 — Debian jessie — Debian Manpages." https://manpages.debian.org/jessie/apache2/a2ensite.8.en.html (accessed Dec. 04, 2020).

[113] "VPN Software Solutions & Services For Business | OpenVPN." https://openvpn.net/ (accessed Dec. 05, 2020).

[114] "/docs/manmaster/man1/openssl.html." https://www.openssl.org/docs/manmaster/man1/openssl.html (accessed Dec. 05, 2020).

[115] "openvpn(8): secure IP tunnel daemon - Linux man page." https://linux.die.net/man/8/openvpn (accessed Dec. 05, 2020).

[116] “minicom(1) - Linux manual page.” https://man7.org/linux/man-pages/man1/minicom.1.html (accessed Dec. 07, 2020).

[117] “sz(1): XMODEM, YMODEM, ZMODEM file send - Linux man page.” https://linux.die.net/man/1/sz (accessed Dec. 07, 2020).

[118] “rz(1): XMODEM, YMODEM, ZMODEM file receive - Linux man page.” https://linux.die.net/man/1/rz (accessed Dec. 07, 2020).

[119] “in.tftpd(8) - Linux man page.” https://linux.die.net/man/8/in.tftpd (accessed Dec. 07, 2020).

[120] “expect(1) - Linux man page.” https://linux.die.net/man/1/expect (accessed Dec. 07, 2020).

[121] “xmllint(1): XML tool - Linux man page.” https://linux.die.net/man/1/xmllint (accessed Nov. 25, 2020).

[122] “DocumentacionUsuario1.1.0/OgAgent – Proyecto OpenGnsys.” https://opengnsys.es/trac/wiki/DocumentacionUsuario1.1.0/OgAgent (accessed Dec. 01, 2020).

[123] “ln(1) - Linux manual page.” https://man7.org/linux/man-pages/man1/ln.1.html (accessed Dec. 02, 2020).

[124] “Add-WindowsCapability (DISM) | Microsoft Docs.” https://docs.microsoft.com/en-us/powershell/module/dism/add-windowscapability?view=win10-ps (accessed Dec. 04, 2020).

[125] “Start-Service (Microsoft.PowerShell.Management) - PowerShell | Microsoft Docs.” https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/start-service?view=powershell-7.1 (accessed Dec. 04, 2020).

[126] “Set-Service (Microsoft.PowerShell.Management) - PowerShell | Microsoft Docs.” https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/set-service?view=powershell-7.1 (accessed Dec. 04, 2020).

[127] “Release Notes for the Catalyst 3750-X and 3560-X Switches, Cisco IOS Release 15.2(4)E - Cisco.” https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-2_4_e/releasenotes/rn-1524e-3750x3560x.html (accessed Dec. 07, 2020).