TRABAJO DE FIN DE TÍTULO

Grado en Ingeniería Informática

VIRTUALIZACIÓN Y OPTIMIZACIÓN DE SISTEMAS DEL INSTITUTO UNIVERSITARIO DE CIENCIAS Y TECNOLOGÍAS CIBERNÉTICAS DE LA ULPGC

Autor

Kilian Marco Luque Romero

Tutores

Dr. Alexis Quesada Arencibia – Ciencias de la Computación e Inteligencia Artificial

Dr. Carmelo Rubén García Rodríguez – Ciencias de la Computación e Inteligencia Artificial

Junio de 2020 – Las Palmas de Gran Canaria

DESCRIPCIÓN DEL DOCUMENTO

Trabajo de Fin de Título en el Grado de Ingeniería Informática, con intensificación en Tecnologías de la Información, de la Universidad de Las Palmas de Gran Canaria presentado por el alumno:

Kilian Marco Luque Romero

TÍTULO DEL PROYECTO

Virtualización y optimización de sistemas del Instituto Universitario de Ciencias y Tecnologías Cibernéticas de la Universidad de Las Palmas de Gran Canaria.

TUTORES

Dr. Alexis Quesada Arencibia
Dr. Carmelo Rubén García Rodríguez

AGRADECIMIENTOS

A mis tutores Dr. Alexis Quesada y Dr. Carmelo Rubén García, por su paciencia, tiempo y dedicación en la resolución de dudas y su completa asistencia en el desarrollo de este proyecto.

A los compañeros integrantes y colaboradores del Instituto Universitario de Ciencias y Tecnologías Cibernéticas por su amabilidad y atención durante mi estancia en el centro. En especial a Jonathan Alemán y Alexis Espino por sus conocimientos, ayuda y guía en el estudio del funcionamiento de la infraestructura del instituto y sus actividades.

A mi familia y amigos, por todo el ánimo, el apoyo y la compañía recibida durante el transcurso del grado y la realización de este trabajo de fin de título.

RESUMEN

El Instituto Universitario de Ciencias y Tecnologías Cibernéticas de la ULPGC presta una serie de servicios web imprescindibles para el desarrollo de proyectos y otras actividades por parte del personal del centro.

Este proyecto consistió en transformar la infraestructura que originalmente daba soporte a esos servicios, compuesta por varios servidores físicos, en una infraestructura optimizada basada en la virtualización. Para ello se aprovechó un solo servidor físico como hipervisor para crear los servidores virtuales en los cuales luego se migraron y desplegaron los servicios web del IUCTC.

Además, se implantó un sistema de monitorización, un sistema automatizado de copias de seguridad y un sistema de alimentación ininterrumpida.

ABSTRACT

The University Institute of Sciences and Cybernetic Technologies of the ULPGC provides a series of essential web services for the development of projects and other activities by the staff of the center.

This project consisted of transforming the infrastructure that originally supported those services, made up of several physical servers, in an optimized infrastructure based on virtualization. To this end, a single physical server was used as a hypervisor to create the virtual servers in which the IUCTC web services were later migrated and deployed.

In addition, a monitoring system, an automated backup system and an uninterrupted power supply system were implemented.

PALABRAS CLAVE

Virtualización, Proxy inverso, Monitorización, Copias de seguridad, Alimentación ininterrumpida

KEYWORDS

Virtualization, Reverse proxy, Monitoring, Backup, Uninterrupted power

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE ILUSTRACIONES

INTRODUCCIÓN

La virtualización es una tecnología que se ha extendido bastante a nivel de grandes corporaciones durante estos últimos años y está siendo utilizada, cada vez más, por empresas de todo tipo, incluso de forma particular.

¿Pero qué es la virtualización?

De manera formal se puede definir la virtualización como la creación de una versión virtual de algún recurso tecnológico, como puede ser una plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento o cualquier otro recurso de red.

Así pues, mediante esta tecnología, se pueden ejecutar de forma virtual múltiples recursos, procesos, o incluso sistemas operativos aislados e independientes entre sí a partir de la carga de trabajo de un mismo servidor físico.

¿Cómo funciona la virtualización?

Siguiendo con su definición la virtualización tiene muchas formas de aplicación, desde la simple partición de un recurso como un disco duro hasta la creación de varias máquinas virtuales con su propio sistema operativo en una misma máquina física.

Esta última forma de virtualización se denomina “virtualización de plataforma” y se puede considerar como la forma de virtualización más típica cuando se habla de virtualización como tal.

El núcleo central de la virtualización de plataforma se denomina “hipervisor” y se lo puede definir como la pieza de software, firmware o hardware que permite la creación y ejecución de máquinas virtuales. Para ello, el hipervisor crea una capa de software que sirve para separar los recursos de la máquina física de los de las máquinas virtuales que se ejecuten.

La máquina en la que se instala el hipervisor se denomina “máquina anfitriona” y las máquinas virtuales que se ejecutan sobre ella, “máquinas huésped”.

Además, según donde esté situado, podemos encontrarnos con dos tipos de hipervisores:

- Tipo 1, nativo, *unhosted* o *bare metal*. Se ejecuta directamente sobre el hardware.
- Tipo 2 o *hosted*. Se ejecuta sobre el sistema operativo.

Dentro de la virtualización de plataforma podemos encontrar distintos tipos de virtualización:

- Virtualización parcial. Se virtualizan múltiples instancias de gran parte (pero no de todo) del entorno subyacente del hardware, particularmente los espacios de direcciones. Este tipo de virtualización permite la compartición de recursos y el alojamiento de procesos, pero no instancias separadas de sistemas operativos en las máquinas huésped.
- Virtualización completa. Se virtualiza el hardware suficiente para permitir que el sistema operativo de una máquina huésped (diseñado para la misma CPU que la de la máquina anfitriona) se ejecute de forma aislada. De esta forma, pueden ejecutarse múltiples instancias al mismo tiempo.
- Emulación. Se simula el hardware suficiente para permitir que el sistema operativo de una máquina huésped (diseñado para una CPU completamente distinta a la de la máquina anfitriona) se ejecute de forma aislada.
- Virtualización a nivel de sistema operativo. Las máquinas huésped comparten el mismo sistema operativo que la máquina física sobre la que se ejecutan y mediante el kernel del propio sistema operativo se implementa el entorno virtual de las mismas. Además, estas máquinas huésped también se ejecutan de forma aislada.
- Paravirtualización. El hipervisor proporciona una API que puede ser utilizada por las máquinas huésped, pero solo mediante la modificación de su sistema operativo. De esta manera se podría optimizar el sistema operativo de dichas máquinas, ya que este pasaría a poder detectar que está siendo ejecutado en un entorno virtual.

¿Qué beneficios reporta la virtualización?

La cantidad de beneficios que puede reportar la utilización de esta tecnología son muchos, por ello vamos a destacar los más notables.

Para empezar, permite reducir la cantidad de servidores físicos requeridos en un centro de procesamiento de datos al poder consolidar varias aplicaciones, entornos de desarrollo y sistemas operativos en un mismo servidor físico, traduciéndose en un menor costo de mantenimiento y una mayor escalibilidad de dichos servidores.

Además, al estar los elementos virtualizados aislados entre sí los posibles accidentes o ataques a dichos elementos no se propagarían a los demás sistemas, significando una mayor seguridad.

Por otra parte, al ser los elementos virtualizados independientes entre sí su administración podría realizarse de forma más sencilla y tendrían un mayor grado de personalización.

Dicho esto, podemos entender por qué esta tecnología es tan utilizada por cada vez más corporaciones y por qué el centro de procesamiento de datos (CPD) del Instituto Universitario de Ciencias y Tecnologías Cibernéticas (en adelante IUCTC) ha decidido sumarse también a la iniciativa.

El presente proyecto se ha realizado con el fin de virtualizar la infraestructura sobre la que operan los distintos servicios web del IUCTC en respuesta a la necesidad de optimizar el funcionamiento de su CPD.

Para ello se ha decidido utilizar un mismo servidor físico del CPD del IUCTC para crear, mediante virtualización completa, distintos servidores virtuales en los que alojar y englobar todos los servicios web del centro. De esta forma se esperaba poder tener en ejecución en la máquina anfitriona varios servidores web virtuales a la vez, alojando distintos servicios web, con distintos entornos de desarrollo e incluso con distintos sistemas operativos.

Adicionalmente, para dotar a dicho servidor de una mayor tolerancia a fallos, así como para asegurar el funcionamiento y la disponibilidad de los servicios web del instituto, se decidió implantar en la infraestructura del CPD un sistema de monitorización, un sistema de copias de seguridad y un sistema alimentación ininterrumpida.

El sistema de monitorización se encargaría de comprobar que tanto el servidor de virtualización como los servicios web alojados en él se encuentren en buen estado y notificaría de cualquier incidente a los administradores del CPD.

El sistema de copias de seguridad se encargaría de realizar copias periódicas de los datos referentes a las distintas aplicaciones y servicios web del IUCTC y las almacenaría en un servidor de copias de seguridad en caso de pérdida de datos.

Por último, el sistema de alimentación ininterrumpida se encargaría de encender y apagar de forma controlada los distintos servidores físicos del CPD del IUCTC en caso de apagón eléctrico.

ESTRUCTURA DEL DOCUMENTO

Este documento se encuentra estructurado en varios capítulos, en los que se detalla el desarrollo de los siguientes apartados:

- Capítulo 1: Estado actual y objetivos

  En este capítulo se realiza una breve descripción sobre el estado original de la infraestructura del IUCTC y se exponen los objetivos del proyecto.

- Capítulo 2: Competencias específicas cubiertas

  En este capítulo se enumeran y justifican las distintas competencias cubiertas en el desarrollo del proyecto.

- Capítulo 3: Aportaciones

  En este capítulo se detectan las distintas aportaciones del proyecto en distintos ámbitos, tales como el técnico y el económico.

- Capítulo 4: Normativa y legislación

  En este capítulo se analizan las licencias de las tecnologías utilizadas y las medidas de seguridad tomadas en el desarrollo del proyecto en cumplimiento con la normativa vigente.

- Capítulo 5: Metodología y planificación

  En este capítulo se expone la metodología de trabajo utilizada y los ajustes de la planificación inicial del desarrollo del proyecto.

- Capítulo 6: Tecnologías utilizadas

  En este capítulo se listan y definen las distintas tecnologías y herramientas empleadas en el desarrollo del proyecto.

- Capítulo 7: Análisis

  En este capítulo se analizan los distintos sistemas, servidores y servicios web que conformaban la infraestructura original del instituto.

- Capítulo 8: Diseño

  En este capítulo se presenta un diseño en forma de esquema de red de la infraestructura ideada para el proyecto, haciendo énfasis en las ventajas de la utilización de un soporte virtual y un *proxy* inverso.

- Capítulo 9: Desarrollo

  En este capítulo se detalla en profundidad cada uno de los procesos realizados para implementar la nueva infraestructura basada en virtualización sin producir alteraciones notables en la disponibilidad de los servicios web del centro.

- Capítulo 10: Evaluación

  En este capítulo se recopilan las distintas comprobaciones de funcionamiento y rendimiento realizadas en los sistemas de la infraestructura implementada.

- Capítulo 11: Conclusiones y trabajos futuros

  En este capítulo se manifiestan las conclusiones sobre el desarrollo del proyecto y las posibles líneas de mejora o ajuste de los sistemas de la infraestructura implementada durante el mismo.

- Anexos

  En este apartado se incluye toda la información complementaria del proyecto.

- Fuentes de información

  En este apartado se incluyen todas las fuentes de información consultadas para el desarrollo del proyecto.

1 CAPÍTULO 1: ESTADO ACTUAL Y OBJETIVOS

1.1 ESTADO ACTUAL

El Instituto Universitario de Ciencias y Tecnologías Cibernéticas se constituye como tal en la Universidad de Las Palmas de Gran Canaria, con carácter interdisciplinario y vocación internacional, haciendo énfasis en la investigación básica y aplicada en ciencia y tecnología de los computadores y la computación, teoría de sistemas, ciencias cognitivas, percepción artificial, biomedicina computacional, neurociencia computacional, economía computacional, tecnologías de la información y robótica [1].

En el momento de comenzar este trabajo, el IUCTC prestaba una serie de servicios web de vital importancia para el desarrollo de las actividades de este, pues eran la pieza fundamental de muchas de las labores y proyectos que ahí llevan a cabo los investigadores, desarrolladores, administradores y demás personal del centro. Además, entre esos servicios se encuentra el propio portal web del instituto mediante el cual las personas ajenas a él pueden conocer los objetivos y las responsabilidades de este.

Dichos servicios operaban sobre una infraestructura que funcionaba correctamente y permitía el cumplimiento de las características que debe tener un servicio de tecnologías de la información (TI): utilidad y garantía. Sin embargo, se detectaron dos aspectos que se podían mejorar.

Por una parte, la infraestructura original que daba soporte a los servicios del IUCTC, si bien es cierto que lo hacía de manera eficaz, no lo hacía eficientemente. Esto se debía, principalmente, al uso de múltiples servidores con un bajo índice de utilización de recursos ya que no se estaba aprovechando toda la capacidad de cada uno.

Por otra parte, se hacía un ligero seguimiento del estado de los servicios del centro, pues los sistemas de monitorización y alimentación ininterrumpida del centro originalmente estaban incompletos, resultando en una cantidad de tiempo de respuesta ante fallos mayor de lo deseada. Relacionado con lo anterior, la infraestructura no estaba respaldada por un sistema automatizado de copias de seguridad por lo que el índice de tolerancia ante fallos también era bajo.

La idea de este proyecto nació de la necesidad de optimizar la infraestructura que daba soporte a todos los servicios del IUCTC y, para ello, se decidió transformarla en una

infraestructura virtual debidamente monitorizada y respaldada que cumpliera de forma más eficiente su propósito original.

1.2 OBJETIVOS

El desarrollo de este proyecto ha implicado la realización del análisis de los sistemas, los servicios y demás recursos de los que se disponía en el IUCTC, así como del estudio, valoración y comparación de las distintas tecnologías que se utilizaron con el fin de encontrar las que más se ajustaran al logro de los siguientes objetivos:

- Diseñar e implementar una nueva infraestructura basada en la virtualización que sirva de soporte a los servicios web del IUCTC.
- Implantar un sistema de copias de seguridad de respaldo de los datos de los servidores principales del IUCTC.
- Implantar un sistema de monitorización que permita disponer el correcto funcionamiento de los servidores y los servicios web del IUCTC.
- Implantar varios sistemas de alimentación ininterrumpida que respalden los servidores físicos del IUCTC en caso de apagón eléctrico.

2 CAPÍTULO 2: COMPETENCIAS ESPECÍFICAS CUBIERTAS

El estudio del Grado en Ingeniería Informática implica la adquisición de unas competencias generales según el RD 1393/2007, unas competencias establecidas por la Universidad de Las Palmas de Gran Canaria para todas sus titulaciones y unas competencias propias del título según lo expuesto en el Anexo II de la Resolución del 8 de julio de 2009 de la Secretaría General de Universidades (BOE del 4 de agosto de 2009), en las que se incluyen unas competencias específicas de intensificación [2].

2.1 COMUNES A LA INGENIERÍA INFORMÁTICA

2.1.1 CII01

"Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente."

Debido a que se ha diseñado e implementado una infraestructura virtual monitorizada y respaldada que se basa en las tecnologías y aplicaciones más utilizadas y mejor conocidas en el mercado que cumplen con los estándares de fiabilidad, seguridad y calidad, así como con la normativa vigente.

2.1.2 CII02

"Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social."

Debido a que ha sido necesario concebir y poner en funcionamiento una nueva infraestructura desde la que se ha tenido que desplegar de nuevo los servicios a los que la infraestructura original daba soporte, valorando su impacto para la organización y asegurando que este fuera positivo.

2.1.3 CII05

"Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas."

Debido a la necesidad de estudiar los distintos sistemas, tecnologías y servicios de la infraestructura original, así como de la nueva, para su correcta administración y mantenimiento.

2.1.4 CII010

"Conocimiento de las características, funcionalidades y estructura de los Sistemas Operativos y diseñar e Implementar aplicaciones basadas en sus servicios."

Debido a que ha sido necesario conocer las características y la estructura de CentOS 7 para la implementación de la infraestructura virtual diseñada: instalación y configuración de KVM, creación y administración de máquinas y redes virtuales, instalación y configuración de XAMPP y NGINX, etc.

2.1.5 CII013

"Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los Sistemas de Información, incluidos los basados en web."

Debido a que ha sido necesario tener conocimiento de las tecnologías de almacenamiento, procesamiento y acceso a los sistemas del IUCTC, tales como MySQL, PHP, phpMyAdmin, SSH, etc. para poder administrar, migrar y desplegar correctamente los servicios del IUCTC.

2.1.6 CII018

"Conocimiento de la normativa y la regulación de la informática en los ámbitos nacional, europeo e internacional."

Debido a que la implantación y el funcionamiento de la nueva infraestructura debe recoger los aspectos señalados en las normativas de ámbito nacional, europeo e internacional.

2.2 TECNOLOGÍAS DE LA INFORMACIÓN

2.2.1 TI01

"Capacidad para comprender el entorno de una organización y sus necesidades en el ámbito de las tecnologías de la información y las comunicaciones."

Debido a que el diseño de la nueva infraestructura virtual y su desarrollo ha implicado elaborar un plan de acción mediante el conocimiento y el estudio de los distintos sistemas, servicios y recursos del IUCTC, así como sus procedimientos y necesidades.

2.2.2 TI02

"Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados."

Debido a que el desarrollo de este proyecto se ha realizado en base a la utilización tanto de los recursos disponibles del IUCTC como de tecnologías de software libre y código abierto que, junto a la ejecución de buenas prácticas, han resultado en una solución de calidad, eficiente, segura y de bajo coste.

2.2.3 TI04

"Capacidad para seleccionar, diseñar, desplegar, integrar y gestionar redes e infraestructuras de comunicaciones en una organización."

Debido a que ha sido necesario diseñar y desplegar una nueva infraestructura basada en la virtualización, lo que ha conllevado también el despliegue y la gestión de una red virtual en la que se comuniquen los distintos servidores virtuales del IUCTC.

2.2.4 TI05

"Capacidad para seleccionar, desplegar, integrar y gestionar sistemas de información que satisfagan las necesidades de la organización, con los criterios de coste y calidad identificados."

Debido a que el desarrollo del proyecto ha dado como resultado una solución de una calidad esperada sin suponer un costo adicional al identificado al comienzo de este.

2.2.5 TI06

"Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil."

Debido a que la creación de la nueva infraestructura ha implicado la creación de servidores web virtuales que operan mediante Internet.

2.2.6 TI07

"Capacidad para comprender, aplicar y gestionar la garantía y seguridad de los sistemas informáticos."

Debido a que ha sido necesario asegurar los distintos sistemas informáticos que componen la nueva infraestructura ya que son sistemas críticos que ofrecen una serie de servicios indispensables para el desarrollo de las actividades del IUCTC y se tratan datos de carácter personal que deben protegerse imprescindiblemente siguiendo la legislación vigente.

2.3 TRABAJO DE FIN DE GRADO

2.3.1 TFG01

"Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas."

Debido a que se ha desarrollado este proyecto de forma individual, partiendo de una idea propia de los tutores, que ha conllevado la realización de distintas fases de trabajo en las que se han utilizado los conocimientos y se han aplicado las competencias propias adquiridas en el grado.

3 CAPÍTULO 3: APORTACIONES

3.1 ENTORNO TÉCNICO

El desarrollo de este proyecto ha implicado un cambio importante en la infraestructura del Instituto Universitario de Ciencias Tecnológicas y Cibernéticas al implementar una nueva infraestructura basada en la virtualización. De esta forma, se ha logrado centralizar todos los servidores y servicios web del centro en un solo servidor físico permitiendo obtener un mejor índice de utilización de recursos.

Además, se implementaron nuevos sistemas de monitorización, alimentación ininterrumpida y copias de seguridad y supusieron una optimización en el índice de tolerancia frente a fallos con respecto a la infraestructura original.

3.2 ENTORNO ECONÓMICO

Al englobar todos los servidores y servicios del centro en un solo servidor físico se redujo el número de servidores físicos que se necesitaban tener en funcionamiento al mismo tiempo. Esto eventualmente supondría una reducción en el consumo de energía, así como en los costes de mantenimiento de los sistemas del centro, dando como resultado un menor gasto económico con respecto a la infraestructura original.

4 CAPÍTULO 4: NORMATIVA Y LEGISLACIÓN

4.1 LICENCIAS SOFTWARE

Una licencia de software [3] es un instrumento legal que, a través del derecho contractual, rige el uso, distribución y modificación de un determinado software. Se trata de un contrato entre dos partes: el autor del software protegido por derechos de autor o licenciante y el usuario final o licenciatario.

Generalmente, una licencia de software otorga al licenciatario permiso para hacer uso de una o más copias de software de forma que, de otro modo, dicho uso podría constituir una violación de los derechos de autor exclusivos del propietario del software.

Para el desarrollo de este proyecto se ha utilizado en su totalidad software libre, por lo que ha sido necesaria la intervención de licencias software.

4.1.1 Licencia GNU GPL

La Licencia Pública General de GNU [4] o más conocida por su nombre en inglés *GNU General Public License* (GNU GPL) es una licencia de derecho de autor ampliamente usada en el mundo del software libre y código abierto, y garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software. Su propósito es doble: declarar que el software cubierto por esta licencia es libre, y protegerlo de intentos de apropiación que restrinjan esas libertades a nuevos usuarios cada vez que la obra es distribuida, modificada o ampliada.

A lo largo de su existencia, esta licencia ha tenido tres versiones: GNU GPLv1 [5], GNU GPLv2 [6] y GNU GPLv3 [7].

Bajo los términos de esta licencia se ha utilizado el siguiente software:

- El sistema operativo CentOS 7 (GPLv2).
- El sistema de gestión de cortafuegos FirewallD (GPLv2).
- Varias partes del módulo de virtualización KVM: el módulo de kernel de KVM (GPLv2) y el emulador de QEMU en modo usuario de Linux (GPLv1).
- El sistema de gestión de bases de datos MariaDB (GPLv2).
- El sistema de monitorización Zabbix (GPLv2).
- El software de gestión de UPS de Riello, PowerShield[3] (GPLv2+).
- El programa de transferencia de ficheros LFTP (GPLv3+).
- El sistema de control de versiones Git (GPLv2).

4.1.2 Licencia GNU LGPL

La Licencia Pública General Reducida de GNU [8] o más conocida por su nombre en inglés *GNU Lesser General Public License* (GNU LGPL) es una licencia de software creada por la *Free Software Foundation* que pretende garantizar la libertad de compartir y modificar el software cubierto por ella, asegurando que el software es libre para todos sus usuarios.

La principal diferencia entre la GPL y la LGPL es que la última puede enlazarse a un programa no-GPL, que puede ser software libre o software no libre.

A lo largo de su existencia, esta licencia también ha tenido tres versiones: GNU LGPLv2 [9], GNU LGPLv2.1 [10] y GNU LGPLv3 [11].

Bajo los términos de esta licencia se ha utilizado el siguiente software:

- Varias partes del módulo de virtualización KVM: el módulo de usuario de KVM (LGPLv2), tanto la librería del núcleo de procesador virtual de QEMU como el emulador de sistema de computador de QEMU (LGPLv2) y los ficheros de BIOS (LGPLv2+).
- El sistema de control de versiones Git (LGPLv2.1).

4.1.3 Licencia BSD

La licencia BSD [12] (*Berkeley Software Distribution*) es una licencia de software libre permisiva que imponen las restricciones mínimas sobre el uso y la distribución de software, en contraste con las licencias *copyleft*. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

Existen tres versiones de esta licencia: la licencia BSD original (de 4 cláusulas) [13], la nueva licencia BSD o licencia BSD modificada (de 3 cláusulas) [14] y la licencia BSD simplificada o licencia FreeBSD (de 2 cláusulas) [15].

En este caso, interviene la **licencia BSD simplificada**, utilizada por el proyecto FreeBSD, en la cual se elimina la última cláusula de la licencia BSD modificada y se agrega un aviso de que las opiniones y puntos de vista de los contribuyentes del proyecto no representan necesariamente la visión del proyecto FreeBSD.

Bajo los términos de la licencia BSD simplificada se ha utilizado el servidor web/*proxy* inverso NGINX.

4.1.4 Licencia Apache

La licencia de Apache [16] es una licencia de software libre permisiva creada por la *Apache Software Foundation* (ASF). La licencia Apache requiere la conservación del aviso de derecho de autor y el descargo de responsabilidad, pero no es una licencia *copyleft*, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas.

A lo largo de su existencia, esta licencia también ha tenido tres versiones: Apache License 1.0 [17], Apache License 1.1 [18] y Apache License 2.0 [19].

En este caso, interviene la **licencia Apache 2.0**. Esta versión fue creada con el objetivo de facilitar su utilización para proyectos que no fueran ASF y mejorar la compatibilidad con software basado en GPL.

Bajo los términos de la versión 2 de esta licencia se ha utilizado el siguiente software:

- El servidor web Apache.
- El servidor XRDP.

4.1.5 Licencia PHP

La licencia PHP [20] es la licencia bajo la que se publica el lenguaje de programación PHP. De acuerdo con la Free Software Foundation es una licencia de software libre no *copyleft* y una licencia de código abierto según la Open Source Initiative. Debido a la restricción en el uso del término "PHP", no es compatible con la licencia GPL.

La versión 3 [21] del lenguaje PHP usa una licencia dual: el código fuente de esa versión se encuentra disponible bajo la licencia PHP y la GNU GPL. A partir de la versión 4 de PHP, esta última dejó de usarse debido a (según los desarrolladores del lenguaje) las restricciones impuestas por el *copyleft* a la reutilización de los contenidos.

Bajo los términos de la versión 3 de esta licencia se ha utilizado el lenguaje de programación PHP.

4.1.6 Tabla resumen de las licencias software utilizadas

A continuación, se muestra una tabla resumen (Tabla 1) con la relación entre las tecnologías utilizadas en este proyecto y sus correspondientes licencias:

	GNU GPL	GNU LGPL	BSD simplificada	Apache 2.0	PHP 3
CentOS 7	X				
FirewallD	X				
XRDP				X	
KVM	X	X			
NGINX			X		
Apache				X	
MariaDB	X				
PHP					X
Zabbix	X				
PowerShield[3]	X				
LFTP	X				
Git	X	X			

Tabla 1: Resumen de las licencias software de las tecnologías utilizadas en el proyecto

4.2 SEGURIDAD DE LOS DATOS

El Reglamento General de Protección de Datos (RGPD) [22] es el reglamento europeo relativo a la protección de las personas físicas en lo que respecta al tratamiento de sus datos personales y a la libre circulación de estos datos. El cumplimiento del RGPD es indispensable hoy en día para cualquier negocio u organización que trate datos de carácter personal dentro de la Unión Europea [23].

Según este reglamento, se debe garantizar la protección de los datos personales que se traten en el desarrollo de este proyecto, especialmente aquellos almacenados en bases de datos.

Para cumplir con lo anterior fue necesario proteger estos datos frente a accesos no autorizados y/o no deseados mediante la ejecución de buenas prácticas en seguridad informática, tanto para la migración como para el despliegue en la nueva infraestructura de los servicios del IUCTC. Por ello, se tomaron las siguientes medidas:

- La protección con usuario y contraseña a nivel de sistema operativo de todos los sistemas tanto físicos como virtuales que han intervenido en el proyecto.
- La protección con usuario y contraseña de todos los accesos por FTP (*File Transfer Protocol*) a todos los sistemas de la nueva infraestructura.

- La protección con usuario y contraseña de todas las bases de datos asociadas a los servicios operativos en la nueva infraestructura.
- La realización de copias de seguridad de todos los datos tratados en el desarrollo de este proyecto.

5 CAPÍTULO 5: METODOLOGÍA Y PLANIFICACIÓN

5.1 METODOLOGÍA

Debido a que se trata de un proyecto estable en el cual la incertidumbre no ha tenido cabida, este se ha podido realizar siguiendo una metodología simple basada en un modelo lineal secuencial donde cada fase se completaba antes de comenzar con la siguiente, siendo las fases las mismas especificadas en la planificación inicial: análisis, diseño, desarrollo y evaluación.

En primer lugar, se realizó un estudio en profundidad de los servidores principales del IUCTC y los servicios a los que estaban dando soporte, así como de los demás sistemas y elementos que componían la infraestructura original. Se analizaron cuidadosamente las posibles tecnologías y vías de desarrollo del proyecto.

A continuación, se elaboró un diseño viable de una nueva infraestructura basada en la virtualización para dar soporte a los servicios del instituto, en el cual también se añadieron los sistemas de monitorización, copias de seguridad y alimentación ininterrumpida.

Para el desarrollo se requirió:

- La instalación de un hipervisor para la creación, instalación y configuración de servidores virtuales.
- La migración de todos los servicios del centro a los servidores virtuales, utilizando un servidor temporal que respaldara el proceso.
- La instalación y configuración de un *proxy* inverso que posibilitara el acceso de los clientes a los servicios web alojados en los servidores web virtuales.
- La implementación de un sistema de alimentación ininterrumpida que controlara el apagado y encendido de los servidores físicos.
- El desarrollo de un sistema automatizado de respaldo de los datos de los servidores virtuales.
- La implementación de un sistema de monitorización de los servicios del centro.

Por último, se realizó una evaluación de los sistemas constituyentes de la infraestructura implementada en este proyecto.

5.2 PLANIFICACIÓN INICIAL DEL PROYECTO

La planificación inicial acordada y seguida para el desarrollo del proyecto constó de cuatro fases.

La primera fase, con una duración estimada de 30 horas, consistió en el análisis de requisitos basado en las necesidades del CPD del IUCTC, así como de su infraestructura y las tecnologías a utilizar a lo largo del proyecto.

Para esta primera fase se realizó un análisis del estado del arte, atendiendo a los requisitos en base a las necesidades del CPD del instituto. Adicionalmente, mediante pequeñas entrevistas y preguntas con el personal y el director del IUCTC se estudió rigurosamente la infraestructura original y se obtuvo la información necesaria, incluyendo las tecnologías a utilizar, para diseñar y desarrollar la nueva infraestructura.

La segunda fase, con una duración estimada de 200 horas, consistió en el diseño y desarrollo de la nueva infraestructura basada en la virtualización, la migración de todos los servicios de la infraestructura original a la nueva y la implementación de un sistema de monitorización, otro de copias de seguridad y otro de alimentación ininterrumpida.

Para esta segunda fase se realizó un pequeño diseño de la nueva infraestructura, tomando como referencia el estudio realizado en la fase anterior. En este diseño se presentó un conjunto de servidores virtuales en los que se alojarían todos los servicios web del centro y que operaban dentro de una única máquina física, que además actuaba como *proxy* inverso con el fin de posibilitar la comunicación de los servicios con el exterior y viceversa. Además, se incluyeron los sistemas de monitorización, copias de seguridad y de alimentación ininterrumpida.

Tras el diseño se realizó la instalación y configuración de todos los sistemas necesarios para el correcto funcionamiento de la nueva infraestructura, tras lo cual se procedió a la migración y el despliegue de todos los servicios del centro.

La tercera fase, con una duración estimada de otras 30 horas, consistió en la evaluación del funcionamiento, disponibilidad y rendimiento de la infraestructura desarrollada y los sistemas que la componen.

La última fase, con una duración estimada de 40 horas, consistió en la realización de la documentación de la memoria de fin de título y los manuales de usuario.

5.3 AJUSTES DE LA PLANIFICACIÓN INICIAL DEL PROYECTO

Durante el proceso tanto de estudio como de desarrollo del proyecto se detectó que este tomaría más tiempo del estimado, sobre todo la fase de migración y el despliegue de servicios en la nueva infraestructura. A causa de esto se tuvo que hacer algunos ajustes a la planificación inicial, los cuales se muestran en la siguiente tabla (Tabla 2):

Fases	Duración estimada (horas)	Tareas	Duración real aproximada (horas)
Estudio previo / Análisis	30	Tarea 1.1: Análisis de requisitos basado en las necesidades del CPD del IUCTC.	50
		Tarea 1.2: Estudio de la infraestructura hardware del IUCTC, recursos disponibles y servicios ofrecidos.	
		Tarea 1.3: Estudio de las tecnologías a utilizar.	
Diseño / Desarrollo / Implementación	200	Tarea 2.1: Diseño de una arquitectura de los servicios ofrecidos por el IUCTC en base a una infraestructura virtual.	250
		Tarea 2.2: Instalación y configuración de los sistemas de la nueva infraestructura virtual.	
		Tarea 2.3: Migración de servicios desde los servidores físicos a la nueva infraestructura.	
		Tarea 2.4: Implantación de un sistema de monitorización que asegure el correcto funcionamiento de los servicios ofrecidos desde la nueva infraestructura.	
		Tarea 2.5: Configuración del apagado/encendido controlado de los servidores en caso de caída/restauración del suministro eléctrico.	
		Tarea 2.6: Automatización de las copias de seguridad de todos los sistemas y servicios.	
Evaluación / Validación / Prueba	30	Tarea 3.1: Prueba y análisis de funcionamiento, disponibilidad y rendimiento de la nueva infraestructura virtual y los servicios montados sobre ella.	40
		Tarea 3.2: Prueba y análisis de funcionamiento del nuevo sistema de monitorización de servicios, así como del control de encendido y apagado controlado de los servidores.	
		Tarea 3.3: Prueba y análisis de funcionamiento del nuevo sistema automatizado de copias de seguridad.	
Documentación / Presentación	40	Tarea 4.1: Desarrollo de la documentación de la memoria del trabajo de fin de título.	80

		Tarea 4.2: Desarrollo de los manuales de usuario.	

Tabla 2: Ajustes de la planificación inicial del proyecto

Este proyecto se ha presentado como solución a cuestiones bastante concretas en cuanto al funcionamiento de la infraestructura del IUCTC. Debido a esto, la consecución de los objetivos planteados ha conllevado el estudio minucioso y detallado de varias tecnologías, así como su precisa aplicación, que a su vez ha implicado una inversión de tiempo mayor al no haberse podido tomar como referencia ningún otro proyecto de la misma índole.

A esto se le suma una gran cantidad de horas tanto en procesos de prueba y error como en procesos de espera, como resultado de la instalación y configuración de sistemas, la migración de servicios, la realización de copias de seguridad, etc., naturalmente ligados al concepto de las tecnologías de la información, en el cual se enmarca este proyecto.

6 CAPÍTULO 6: TECNOLOGÍAS UTILIZADAS

6.1 TECNOLOGÍAS PRINCIPALES

6.1.1 CentOS 7

CentOS 7 (*Community ENTerprise Operating System 7*) [24] es un sistema operativo de código abierto, basado en la distribución Red Hat Enterprise Linux, operándose de manera similar, y cuyo objetivo es ofrecer al usuario un software de "clase empresarial" gratuito.

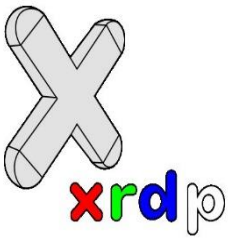
En este caso, se hizo uso de la versión 7 de este sistema al ser la más reciente en el periodo en el que se desarrolló este proyecto.

6.1.2 FirewallD

FirewallD [25] es una herramienta de gestión de cortafuegos para sistemas operativos Linux. Proporciona funciones de cortafuegos actuando como una interfaz para el "framework" del kernel de Linux llamado "Netfilter" (encargado de interceptar y manipular paquetes de red) mediante comandos de "iptables" y actuando como alternativa al servicio con este mismo nombre.

Es la herramienta de gestión de cortafuegos por defecto de CentOS 7.

6.1.3 XRDP



XRDP (*Remote Desktop Protocol for Unix-like operating systems*) [26] es un protocolo de escritorio remoto que permite a los sistemas operativos distintos de Microsoft Windows (como los basados en Unix) proporcionar una experiencia de conexión por escritorio remoto compatible con RDP.

Entendemos por RDP [27] al protocolo de escritorio remoto desarrollado por Microsoft para la conexión mediante interfaz gráfica desde un ordenador (cliente RDP) a otro (servidor RDP) a través de la red.

6.1.4 KVM

KVM (*Kernel-based Virtual Machine* o máquina virtual basada en el núcleo) [28] es un módulo de virtualización en el núcleo de Linux que permite al núcleo funcionar como un hipervisor. KVM requiere de un procesador con extensiones hardware de virtualización, como Intel VT o AMD-V. Este módulo es software libre en su totalidad y ofrece virtualización asistida por hardware para una amplia variedad de sistemas operativos, así como el soporte para la paravirtualización en máquinas virtuales con ciertos sistemas operativos tal que Linux o Windows.

Una de las características de KVM que ha destacado en el desarrollo de este proyecto es el *overcommit*, que es el uso de memoria excediendo aún la memoria física del host.

6.1.5 NGINX

NGINX (pronunciado como "engine-ex") [29] es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3). Es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada. Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS X, etc.) y Windows. El sistema es usado por una larga lista de sitios web conocidos, como: WordPress, Netflix, Hulu, GitHub, Ohloh, SourceForge, TorrentReactor y partes de Facebook (como el servidor de descarga de archivos zip pesados).

Las características básicas [30] de este servidor web son las siguientes:

- Servidor de archivos estáticos, índices y auto-indexado.
- Proxy inverso con opciones de caché.
- Balanceo de carga.
- Tolerancia a fallos.
- Soporte de HTTP y HTTP2 sobre SSL.
- Soporte para FastCGI con opciones de caché.
- Servidores virtuales basados en nombre y/o en dirección IP.
- Streaming de archivos FLV y MP4.
- Soporte para autenticación.

- Compatible con IPv6.
- Soporte para protocolo SPDY.
- Compresión gzip.
- Habilitado para soportar más de 10.000 conexiones simultáneas.

6.1.6 XAMPP

XAMPP [31] es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y PERL. El nombre es en realidad un acrónimo: X (para cualquiera de los diferentes sistemas operativos), Apache, MariaDB/MySQL, PHP, Perl. Además, incluye el servidor de FTP (*File Transfer Protocol*) o protocolo de transferencia de archivos, ProFTPd.

6.1.7 Zabbix

Zabbix [32] es un sistema de monitorización de redes diseñado para monitorizar y registrar el estado de varios servicios de red, servidores y dispositivos de red.

Utiliza MySQL, PostgreSQL, SQLite, Oracle o IBM DB2 como base de datos. Su *backend* está escrito en lenguaje C y el *frontend* web está escrito en PHP.

Zabbix ofrece varias opciones de monitorización:

- Chequeos simples que pueden verificar la disponibilidad y el nivel de respuesta de servicios estándar como SMTP o HTTP sin necesidad de instalar ningún software sobre la máquina monitorizada.
- Un agente Zabbix puede también ser instalado sobre máquinas UNIX y Windows para monitorizar estadísticas como la carga de CPU, la utilización de red, el espacio en disco, etc.
- Como alternativa a instalar el agente sobre las máquinas, Zabbix incluye soporte para monitorizar vía protocolos SNMP, TCP y ICMP, así como también sobre IPMI, JMX, SSH, Telnet y usando parámetros de configuración personalizados.

Zabbix soporta una variedad de mecanismos de notificación en tiempo real, incluyendo XMPP.

6.1.8 ULPnet

ULPnet [33] es la propia red integral de la ULPGC.

Se ha utilizado en este proyecto como herramienta para la administración (creación y eliminación) de dominios de Internet asociados a los distintos servicios web del IUCTC.

6.2 OTRAS HERRAMIENTAS

6.2.1 LFTP

LFTP [34] es un programa cliente de línea de comandos para varios protocolos de transferencia de ficheros. Este programa puede transferir ficheros mediante FTP, FTPS, HTTP, HTTPS, FISH, SFTP, BitTorrent y FTP sobre un proxy HTTP.

Entre las características de LFTP se encuentran las colas de transferencia, la transferencia de ficheros segmentados, la reanudación de descargas parciales, la limitación del ancho de banda y la copia recursiva de directorios.

El cliente se puede utilizar de forma interactiva o automatizada, con el uso de *scripts*. Además, tiene un control de procesos basado en la consola de Unix y una función para programar el momento de ejecución de las transferencias de ficheros.

6.2.2 Git

Git [35] es un software de control de versiones diseñado para promover la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

6.2.3 PowerShield³

PowerShield³ [36] es un software concebido para una eficaz e intuitiva gestión de los sistemas de alimentación ininterrumpida o UPS de Riello. Permite visualizar la información relevante acerca de los sistemas como la tensión de entrada, la carga aplicada, la capacidad de las baterías, etc.

En caso de incidencia, ofrece información detallada sobre el estado del UPS.

La arquitectura cliente/servidor lo convierte en una herramienta ideal para la gestión de sistemas de red multiplataforma.

6.2.4 PIGZ

PIGZ (*Parallel Implementation of GZIP*) [37] es un reemplazo completamente funcional de GZIP que explota al máximo múltiples procesadores y núcleos a hora de comprimir datos.

A continuación, se expone una tabla comparativa de los parámetros de compresión de un fichero del mismo tamaño comprimido utilizando distintos métodos (Ilustración 1).

method	compression level	compress time	compress speed	decompress time	decompress speed	compress cpu %	compress max mem	decompress cpu %	decompress max mem	compression ratio	compress size	original size
zip level 1	1	2.23	91	1.20	168	100	1248	100	1884	2.7396	77366037	211957760
zip level 2	2	2.41	84	1.16	174	99	1248	99	1884	2.8218	75114133	211957760
zip level 3	3	3.07	66	1.13	179	99	1248	100	1884	2.9015	73049991	211957760
zip level 4	4	3.18	64	1.14	177	99	1240	99	1880	2.9829	71057350	211957760
zip level 5	5	4.37	46	1.12	180	99	1236	100	1876	3.0650	69153753	211957760
zip level 6	6	6.26	32	1.10	184	99	1236	99	1876	3.1067	68224573	211957760
zip level 7	7	7.63	26	1.10	184	99	1236	99	1880	3.1197	67939852	211957760
zip level 8	8	11.70	17	1.10	184	99	1232	100	1876	3.1311	67692927	211957760
zip level 9	9	15.07	13	1.11	182	99	1236	100	1876	3.1331	67650352	211957760
gzip level 1	1	3.35	60	1.38	146	99	880	99	748	2.6978	78564491	211957760
gzip level 2	2	3.51	58	1.34	151	99	876	100	748	2.7783	76290028	211957760
gzip level 3	3	4.12	49	1.32	153	100	876	100	748	2.8549	74243077	211957760
gzip level 4	4	4.35	46	1.32	153	100	872	99	748	2.9328	72270908	211957760
gzip level 5	5	5.58	36	1.30	155	99	868	99	744	3.0101	70413441	211957760
gzip level 6	6	7.39	27	1.28	158	99	868	100	744	3.0510	69470340	211957760
gzip level 7	7	8.83	23	1.31	154	99	868	100	748	3.0637	69183113	211957760
gzip level 8	8	12.93	16	1.27	159	99	868	100	748	3.0747	68934529	211957760
gzip level 9	9	16.21	12	1.27	159	99	868	99	748	3.0767	68890507	211957760
bzip2 level 1	1	13.25	15	4.42	46	99	1680	99	884	3.5044	60482475	211957760
bzip2 level 2	2	13.18	15	4.58	44	99	2472	100	1416	3.6460	58134303	211957760
bzip2 level 3	3	13.52	15	4.63	44	99	3264	99	1680	3.7220	56946011	211957760
bzip2 level 4	4	13.76	15	4.80	42	99	4056	100	2208	3.7676	56257778	211957760
bzip2 level 5	5	14.10	14	4.72	43	99	4848	100	2472	3.8013	55758116	211957760
bzip2 level 6	6	14.26	14	4.68	43	99	5640	99	3000	3.8258	55401871	211957760
bzip2 level 7	7	14.46	14	4.45	45	99	6432	100	3196	3.8516	55030762	211957760
bzip2 level 8	8	14.29	14	4.53	45	99	7156	100	3728	3.8706	54760488	211957760
bzip2 level 9	9	14.69	14	4.89	41	99	7880	99	4056	3.8878	54518436	211957760
pigz level 1	1	0.64	316	0.67	302	699	9384	141	1044	2.7087	78248449	211957760
pigz level 2	2	0.67	302	0.66	306	711	9744	140	1044	2.7905	75956899	211957760
pigz level 3	3	0.78	259	0.63	321	731	9588	140	1044	2.8704	73841051	211957760
pigz level 4	4	0.84	241	0.66	306	741	9604	144	1044	2.9415	72057284	211957760
pigz level 5	5	1.07	189	0.65	311	755	9096	143	1044	3.0189	70208849	211957760
pigz level 6	6	1.38	146	0.63	321	770	9496	147	1044	3.0600	69266646	211957760
pigz level 7	7	1.61	126	0.64	316	777	9708	146	1044	3.0726	68982551	211957760
pigz level 8	8	2.30	88	0.63	321	783	9784	146	1044	3.0837	68733630	211957760
pigz level 9	9	2.80	72	0.62	326	775	9888	148	1044	3.0857	68688862	211957760
pbzip2 level 1	1	2.66	76	0.80	253	776	34792	775	32188	3.4992	60572458	211957760
pbzip2 level 2	2	2.72	74	0.82	247	775	37180	780	35372	3.6309	58376006	211957760
pbzip2 level 3	3	2.86	71	0.92	220	775	43852	752	40832	3.7137	57073749	211957760
pbzip2 level 4	4	3.01	67	1.00	202	770	49024	782	42556	3.7401	56670598	211957760
pbzip2 level 5	5	3.14	64	1.11	182	773	54772	784	48864	3.7899	55926672	211957760
pbzip2 level 6	6	3.16	64	1.15	176	777	59904	781	55920	3.7917	55899269	211957760
pbzip2 level 7	7	3.34	61	1.22	166	777	64052	775	58576	3.8035	55725702	211957760
pbzip2 level 8	8	3.67	55	1.40	144	761	70248	781	60460	3.8275	55377413	211957760
pbzip2 level 9	9	3.90	52	1.44	140	765	78064	786	66344	3.8781	54654465	211957760
lbzip2 level 1	1	1.80	112	0.56	361	778	12608	756	50384	3.5004	60551707	211957760
lbzip2 level 2	2	1.72	118	0.60	337	783	17528	775	62540	3.6457	58139060	211957760
lbzip2 level 3	3	1.76	115	0.72	281	783	24400	771	66040	3.7161	57036620	211957760
lbzip2 level 4	4	1.77	114	0.91	222	782	28620	758	62452	3.7650	56296697	211957760
lbzip2 level 5	5	1.83	110	1.03	196	783	41080	780	67492	3.7987	55796067	211957760
lbzip2 level 6	6	1.87	108	1.14	177	781	47148	774	67372	3.8258	55401943	211957760
lbzip2 level 7	7	1.99	102	1.25	162	780	46008	785	70732	3.8458	55113599	211957760
lbzip2 level 8	8	1.99	102	1.33	152	779	51292	776	70068	3.8661	54824122	211957760
lbzip2 level 9	9	1.97	103	1.41	143	783	59764	774	76840	3.8784	54649587	211957760

Ilustración 1: Comparativa de resultados de compresión de distintas implementaciones de GZIP

Se optó por utilizar PIGZ ya que, como se puede observar en la ilustración, tiene una ratio de compresión de ficheros bastante aceptable utilizando el menor tiempo de compresión de la comparativa al nivel medio de compresión. Aunque LBZIP2 a la larga ofrece los mejores resultados, para el desarrollo de este proyecto fue necesario utilizar una herramienta de compresión que priorizara el tiempo de ejecución como lo hace PIGZ.

7 CAPÍTULO 7: ANÁLISIS

7.1 ESTUDIO DE LA INFRAESTRUCTURA ORIGINAL

7.1.1 Previsualización del estado de la infraestructura original

Para empezar, se estudió un documento que fue provisto por el director del IUCTC en el cual se recogía información sobre el estado de los servidores y otros recursos de la infraestructura del centro en ese momento. Este mismo documento se estuvo utilizando a medida que se desarrollaba el proyecto para mantener actualizada dicha información.

El documento contenía información de las máquinas físicas que prestaban o habían prestado algún servicio del IUCTC. Dicha información hacía referencia a:

- Nombres de las máquinas.
- Direcciones IP.
- Direcciones MAC o direcciones físicas de los dispositivos de red.
- Modelos y capacidades de los procesadores.
- Capacidades de almacenamiento.
- Capacidades de memoria.
- Sistemas operativos instalados.
- Herramientas instaladas.
- Usuarios y contraseñas de acceso a las máquinas.
- Usuarios y contraseñas de acceso a los servicios prestados.

7.1.2 Estudio de los servidores y otros recursos de la infraestructura original

Para contrastar la actualidad de los datos expuestos en el apartado anterior se hicieron varias comprobaciones en las máquinas mentadas.

Para ello, en primer lugar, se accedió como administrador a cada una de dichas máquinas y se comprobó su sistema operativo, así como su versión, y se concluyó que estos se basaban en Linux, concretamente en distribuciones como Ubuntu o CentOS 7.

El resto de las comprobaciones realizadas en cada máquina estudiada se exponen en el Anexo I de este documento.

Con los resultados obtenidos de dichas comprobaciones se verificó y se realizó una primera actualización de los datos del documento recibido sobre el estado de los servidores del centro.

Se presenta a continuación un resumen de estos, incluyendo un nombre ficticio con el que se denominará a cada máquina a partir de ahora, un resumen de su función y sus especificaciones básicas:

- Máquina MC. Actuaba como servidor tanto de la Intranet como de la *Wiki* del IUCTC. Se comprobó que la máquina estaba operativa, así como los servicios ofrecidos. A continuación, sus especificaciones básicas (Tabla 3):

Procesador	Intel® Xeon ® Processor E5335 2.00GHz
Número total de procesadores	8
Capacidad de almacenamiento (HDD)	275GB
Capacidad de memoria (RAM)	8GB

Tabla 3: Especificaciones básicas de la máquina MC

- Máquina WN. Actuaba como servidor web de un proyecto del IUCTC. Se comprobó que la máquina estaba operativa, así como el servicio ofrecido. A continuación, sus especificaciones básicas (Tabla 4):

Procesador	Intel® Xeon ® Processor E5335 2.00GHz
Número total de procesadores	8
Capacidad de almacenamiento (HDD)	280GB
Capacidad de memoria (RAM)	4GB

Tabla 4: Especificaciones básicas de la máquina WN

- Máquina MD. Actuaba como servidor web del *Moodle* del IUCTC. Se comprobó que la máquina estaba operativa, así como el servicio ofrecido. A continuación, sus especificaciones básicas (Tabla 5):

Procesador	Intel® Xeon ® Processor E5335 2.00GHz
Número total de procesadores	8
Capacidad de almacenamiento (HDD)	136GB
Capacidad de memoria (RAM)	8GB

Tabla 5: Especificaciones básicas de la máquina MD

- Máquina NG. Actuaba como servidor *Subversion* del IUCTC. Se comprobó que la máquina estaba operativa pero no el servicio ofrecido. A continuación, sus especificaciones básicas (Tabla 6):

Procesador	Intel® Xeon ® Processor 5110 1.60GHz
Número total de procesadores	2
Capacidad de almacenamiento (HDD)	250GB
Capacidad de memoria (RAM)	8GB

Tabla 6: Especificaciones básicas de la máquina NG

- Máquinas B1 y B2. Se utilizaban como servidores de copia de seguridad del IUCTC. Eran máquinas de la marca Buffalo y utilizaban sistemas RAID (*Redundant Array of Independent Disks*). Se comprobó que las máquinas no estaban operativas. A continuación, sus especificaciones básicas (Tablas 7 y 8):

Nivel del sistema RAID	1+0
Número de discos	4
Capacidad de almacenamiento (HDD)	4TB

Tabla 7: Especificaciones básicas de la máquina B1

Nivel del sistema RAID	1
Número de discos	2
Capacidad de almacenamiento (HDD)	2TB

Tabla 8: Especificaciones básicas de la máquina B2

- Máquina MR. Actuaba como servidor web del portal web, entre otros proyectos, del propio IUCTC. Se comprobó que la máquina estaba operativa, así como los servicios ofrecidos. A continuación, sus especificaciones básicas (Tabla 9):

Procesador	Intel® Xeon ® Processor E5649 2.53GHz
Número total de procesadores	24
Capacidad de almacenamiento (HDD)	1.2GB SAS + 1.6TB SATA
Capacidad de memoria (RAM)	64GB

Tabla 9: Especificaciones básicas de la máquina MR

- Máquina NB. Era de tipo NAS (*Network-Attached Storage*) y actuaba como servidor de almacenamiento del IUCTC. Se comprobó que la máquina estaba operativa. A continuación, sus especificaciones básicas (Tabla 10):

Procesador	Intel® Xeon ® Processor 3.06GHz
Número total de procesadores	4
Capacidad de almacenamiento (HDD)	1667GB
Capacidad de memoria (RAM)	1GB

Tabla 10: Especificaciones básicas de la máquina NB

- Máquina NI. Actuaba como servidor de monitorización *Nagios* del IUCTC. Se comprobó que la máquina no estaba operativa y, por lo tanto, tampoco el servicio ofrecido. A continuación, sus especificaciones básicas (Tabla 11):

Procesador	Intel® Core™ 2 Quad Processor Q9550 2.83GHz
Número total de procesadores	4
Capacidad de almacenamiento (HDD)	1296GB
Capacidad de memoria (RAM)	4GB

Tabla 11: Especificaciones básicas de la máquina NI

- Varios sistemas de alimentación ininterrumpida de la marca Riello UPS.

7.1.3 Estudio de los servicios web de la infraestructura original

Para el análisis de los servicios web ofrecidos por el IUCTC se hicieron pequeñas entrevistas y preguntas al personal y, sobre todo, al director, de las cuales se obtuvo conocimiento de la existencia de:

- Un servicio web que operaba como portal web del instituto. Se confirmó que este servicio estaba en funcionamiento.
- Un servicio web que operaba como Moodle del instituto. Este tenía una doble función: por un lado, servir como sistema de e-learning para cursos ofrecidos por el instituto y, por otro, como herramienta de gestión de aprendizaje para los grupos de investigación y para el propio equipo directivo del IUCTC. Se confirmó que este servicio estaba en funcionamiento. A partir de ahora se le denominará "Moodle" a este servicio.
- Un servicio web que operaba como Intranet a la vez que *Wiki* del instituto. Se confirmó que este servicio estaba en funcionamiento, pero estaba obsoleto.
- Varios servicios web asociados a distintos proyectos que estaban siendo llevados a cabo en el instituto. Se confirmó que estos servicios estaban en funcionamiento por el mero hecho de que aún estaban en desarrollo en el momento de hacer el estudio.

Para verificar esta información se accedió como administrador a cada uno de los servidores y se realizó una serie de comprobaciones que también permitirían asociar cada servicio al servidor en el que estaba desplegado. Estas comprobaciones se exponen en el Anexo II de este documento.

Las conclusiones obtenidas de dichas comprobaciones fueron las siguientes:

- En la máquina MC se encontraban desplegados los servicios web de la Intranet y la *Wiki* del centro. Estos servicios no se estaban utilizando en el momento de hacer este estudio.
- En la máquina WN se encontraba desplegado un solo servicio web asociado a un proyecto en preproducción.
- En la máquina MD se encontraba desplegado el servicio web de Moodle del centro.
- En la máquina MR se encontraba desplegado el portal web del centro. Además, se encontraban desplegados en este un total de doce servicios web asociados cada uno a un proyecto distinto en preproducción.

Para ilustrar esta información se ha realizado la siguiente tabla resumen (Tabla 12):

Máquina	Servicios Web	Estado	Prioridad
MR	Portal web del centro 12 proyectos en preproducción	Operativos	Alta
MD	Moodle del centro	Operativos	Alta
WN	1 proyecto en preproducción	Operativos	Media
MC	Intranet del centro *Wiki* del centro	Operativos, obsoletos	Baja

Tabla 12: Información sobre los servicios web de la infraestructura original

Además, cada uno de estos servicios, excepto el de Intranet y *Wiki*, tenía al menos un nombre dominio asociado.

7.2 ANÁLISIS DEL RENDIMIENTO DE LOS SERVIDORES FÍSICOS Y LA DISPONIBILIDAD DE LOS SERVICIOS WEB DE LA INFRAESTRUCTURA ORIGINAL

7.2.1 Análisis del rendimiento de los servidores físicos de la infraestructura original

Se realizó un estudio sobre el rendimiento de los servidores físicos que estaban ofreciendo algún tipo de servicio en la infraestructura original, en este caso, los servidores MC, WN, MD y MR.

Para ello se desarrolló y se programó la ejecución de un *script* denominado *top.sh* que monitorizaría la utilización de los recursos de dichos servidores a lo largo del tiempo. Este procedimiento se encuentra detallado en el Anexo III de este documento.

Para representar de una forma entendible y significativa la información obtenida a partir de dicho *script* se realizaron una serie de gráficos con los resultados obtenidos del fichero */usr/local/bin/SoftMon/top.txt* durante el primer mes de desarrollo del proyecto. Para ello, se hizo un promedio del porcentaje de uso de los recursos de cada servidor durante cada día, desvelando así el problema del bajo índice de utilización de los servidores MC, WN, MD y MR de la infraestructura original.

El gráfico siguiente muestra el índice de utilización de los procesadores de los servidores MC, WN, MD y MR de la infraestructura original durante el primer mes de desarrollo del proyecto. Se puede observar como el porcentaje de uso de procesador de los servidores MC, WN y MD era muy próximo al 0%, mientras que el porcentaje de uso del procesador del servidor MR oscilaba entre el 10% y el 25%, debido a que en este último se alojaba la mayor parte de los servicios web del centro (Ilustración 2).

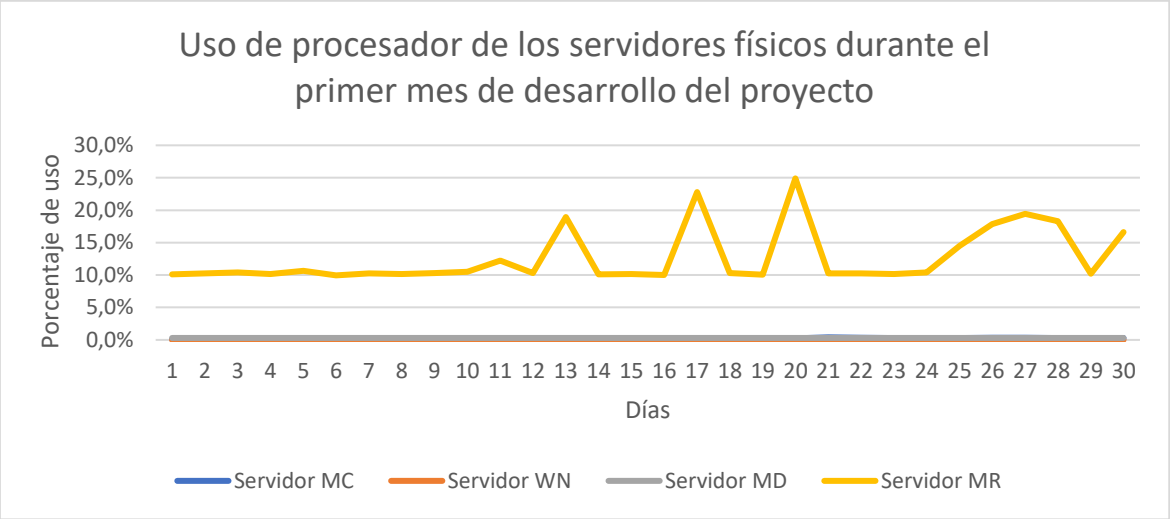


Ilustración 2: Gráfico del uso de procesador de los servidores físicos durante el primer mes de desarrollo del proyecto

El gráfico siguiente muestra índices de utilización bastante diferentes de memoria RAM de los servidores MC, WN, MD y MR de la infraestructura original durante el primer mes de desarrollo del proyecto. Cabe destacar que el porcentaje de uso de la capacidad de memoria RAM del servidor MR era muy próximo al 0% hasta el día 24, a partir del cual se comenzó a realizar multitud de operaciones sobre dicho servidor (Ilustración 3).

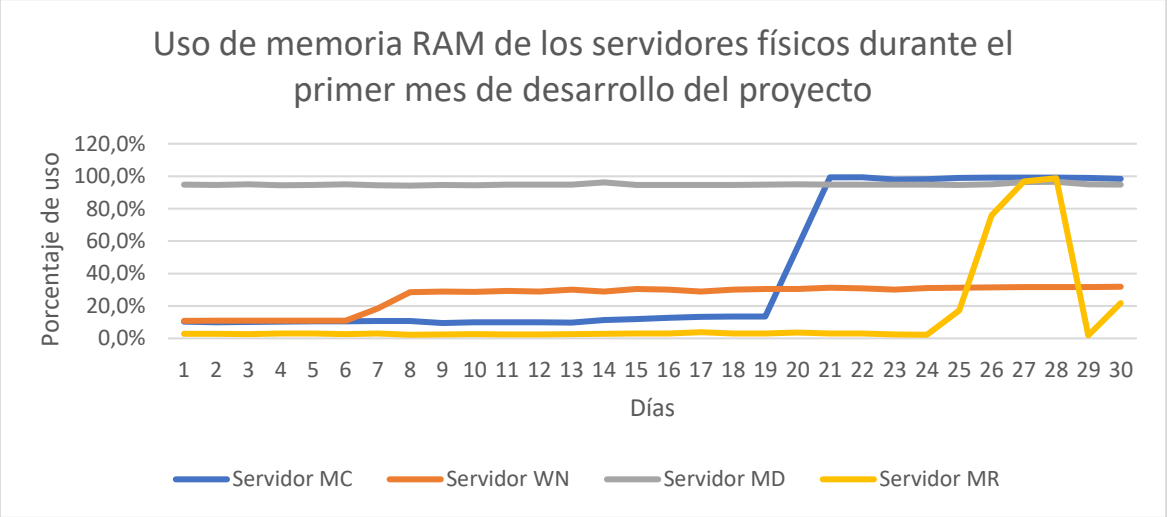


Ilustración 3: Gráfico del uso de memoria RAM de los servidores físicos durante el primer mes de desarrollo del proyecto

El gráfico siguiente muestra índices de utilización generalmente bajos de memoria SWAP de los servidores MC, WN, MD y MR de la infraestructura original durante el primer mes de desarrollo del proyecto. Sin embargo, se puede observar que el porcentaje de uso de la memoria SWAP del servidor MR era muy próximo al 0% hasta el día 25, comportamiento que coincide con el mostrado en el gráfico del porcentaje de uso de la memoria RAM (Ilustración 4).

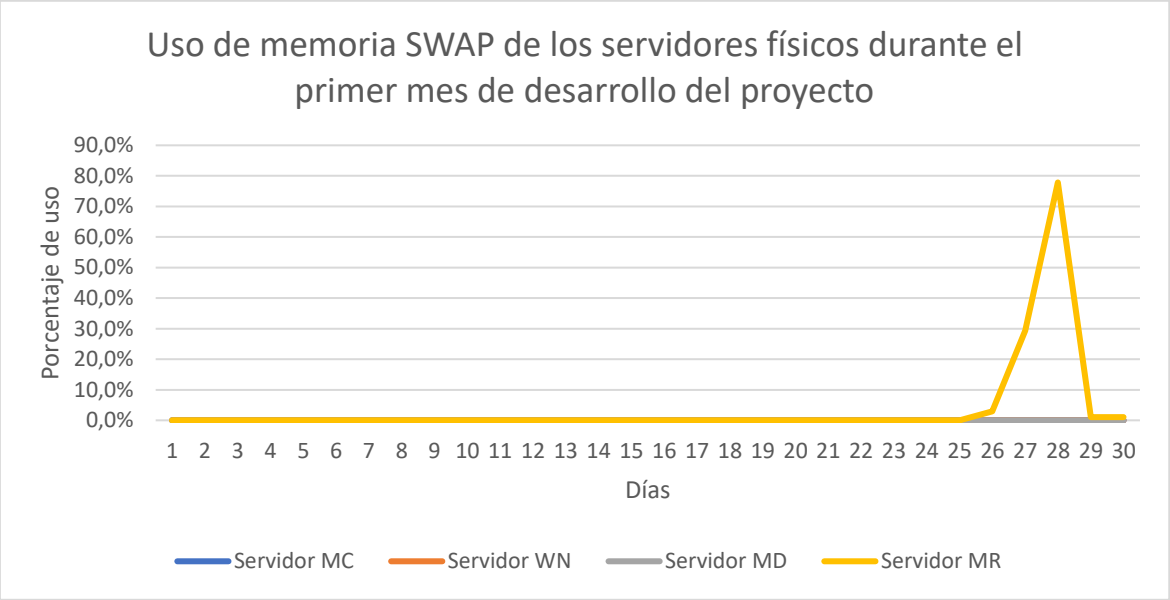


Ilustración 4: Gráfico del uso de memoria SWAP de los servidores físicos durante el primer mes de desarrollo del proyecto

7.2.2 Análisis de disponibilidad de los servicios web de la infraestructura original

Para comprobar la disponibilidad de los servicios web simplemente se accedió mediante un navegador web a cada uno de ellos utilizando su nombre de dominio asociado y comprobando que respondiera correctamente en el tiempo esperado.

8 CAPÍTULO 8: DISEÑO

Para el diseño de la nueva infraestructura se llegó a la conclusión de que lo ideal era trasladar cada uno de los proyectos de los servidores web originales a un servidor web virtual distinto ejecutándose en un mismo servidor físico, que cumpliría la función tanto de hipervisor como de *proxy* inverso, con el objetivo de dar soporte y accesibilidad a todos los servicios web del centro.

Para ello sería fundamental tanto el uso de un *software* de virtualización para la creación y configuración de servidores virtuales como el uso de un *proxy* inverso para permitir a los clientes acceder a los servicios web alojados en dichos servidores.

8.1 VENTAJAS DE LA UTILIZACIÓN DE UN SOPORTE VIRTUAL

Las ventajas más destacadas de la implementación de un diseño centralizado basado en la virtualización serían, para este caso, las siguientes:

- **Mayor índice de utilización de recursos**, al tener varios servidores virtuales aprovechando los recursos de un mismo servidor físico, sobre todo los del procesador y la memoria.
- **Gran ahorro de energía**, al unificar todos los servicios y servidores web en un solo servidor físico. Esto permite a los administradores del centro tener operativos menos servidores físicos o utilizarlos para otros fines, resultando igualmente en una mejora de la productividad.
- **Diversidad y personalización de los entornos de desarrollo,** al haber un solo servidor virtual por cada proyecto web. Esto permite que los creadores, administradores y desarrolladores del centro puedan crear, configurar y gestionar sus propios servidores virtuales para adaptarlos a las necesidades de sus proyectos.

Además, se obtendrían otras ventajas [38] como:

- Consolidación de recursos. La virtualización permite consolidar múltiples recursos de TI. Además de la consolidación de almacenamiento, la virtualización proporciona una oportunidad para consolidar la arquitectura de sistemas, infraestructura de aplicación, datos y base de datos, interfaces, redes, escritorios, e incluso procesos de negocios, resultando en ahorros de costes y en mayor eficiencia.

- Ahorro de espacio. La virtualización permite consolidar muchos sistemas virtuales en menos sistemas físicos.
- Recuperación de desastre/continuidad del negocio. La virtualización puede incrementar la disponibilidad de los índices del nivel de servicio en general y proporcionar nuevas opciones de soluciones para la recuperación de desastre. Hasta el 85% de mejora en tiempo de recuperación de paradas imprevistas.
- Rápida incorporación de nuevos recursos para los servidores virtualizados.
- Administración global centralizada y simplificada.
- Permite gestionar el CPD como un *pool* de recursos o agrupación de toda la capacidad de procesamiento, memoria, red y almacenamiento disponible en la infraestructura.
- Mejora en los procesos de clonación y copia de sistemas. Mayor facilidad para crear entornos de prueba que permiten poner en marcha nuevas aplicaciones sin impactar a la producción, agilizando el proceso de las pruebas.
- Aislamiento. Un fallo general de sistema de una máquina virtual no afecta al resto de máquinas virtuales.
- Reducción de los tiempos de parada.
- Migración en caliente de máquinas virtuales (sin pérdida de servicio) de un servidor físico a otro, eliminando la necesidad de paradas planificadas por mantenimiento de los servidores físicos.
- Contribución al medio ambiente (*Green IT*) por menor consumo de energía en servidores físicos.
- Alta disponibilidad.

8.2 VENTAJAS DE LA UTILIZACIÓN DE UN *PROXY* INVERSO

El despliegue de servidores web virtuales requeriría, en este caso, la utilización de un *proxy* inverso en base a la siguiente propiedad:

"Los proxies inversos pueden operar dondequiera que múltiples servidores web deban ser accesibles a través de una única dirección IP pública. Los servidores web escuchan en diferentes puertos en la misma máquina, con la misma dirección IP local o, posiblemente, en diferentes máquinas y diferentes direcciones IP locales."

Cuya adaptación a este caso sería:

"El proxy inverso operará en el servidor hipervisor de forma que múltiples servidores web virtuales serán accesibles a través de los nombres de dominio asociados a su dirección IP pública."

Además, la utilización de un *proxy* inverso tendría o podría tener las siguientes ventajas [39]:

- Los *proxies* inversos pueden esconder la existencia y características de un servidor de origen o servidores.
- Las funciones de cortafuegos de aplicaciones pueden proteger contra ataques comunes basados en web, como DoS o DDoS. Sin un *proxy* inverso, por ejemplo, puede resultar difícil eliminar programas maliciosos.
- En el caso de sitios web seguros, un servidor web puede no ejecutar la encriptación SSL por sí mismo, sino que encarga esta tarea a un *proxy* inverso que puede estar equipado con hardware de aceleración SSL.
- Un *proxy* inverso puede distribuir la carga de solicitudes entrantes a varios servidores, con cada servidor ejecutando su propia área de aplicación.
- Un *proxy* inverso puede reducir carga de sus servidores de origen mediante el uso de caché web de contenido estático, así como contenido dinámico.
- Un *proxy* inverso puede optimizar el contenido comprimiéndolo para acelerar los tiempos de carga.
- En una técnica llamada *spoon-feed* se puede producir una página generada dinámicamente de una sola vez y servirla al *proxy* inverso, el cual puede devolverla al cliente poco a poco. El programa que genera la página no necesita permanecer

abierto, liberando recursos del servidor durante el tiempo extendido que el cliente requiere para completar la transferencia.

- El *proxy* inverso analiza cada solicitud entrante y la envía al servidor correcto dentro de la red de área local.
- Los *proxies* inversos pueden realizar pruebas A/B y pruebas multivariadas sin colocar etiquetas o código JavaScript en las páginas.
- Un *proxy* inverso puede agregar autenticación de acceso HTTP básico a un servidor web que no tenga ninguna autenticación.

8.3 ESQUEMA DE RED DE LA INFRAESTRUCTURA IDEADA

A continuación, se muestra el esquema de red de la infraestructura virtual ideada para la realización de este proyecto (Ilustración 5).

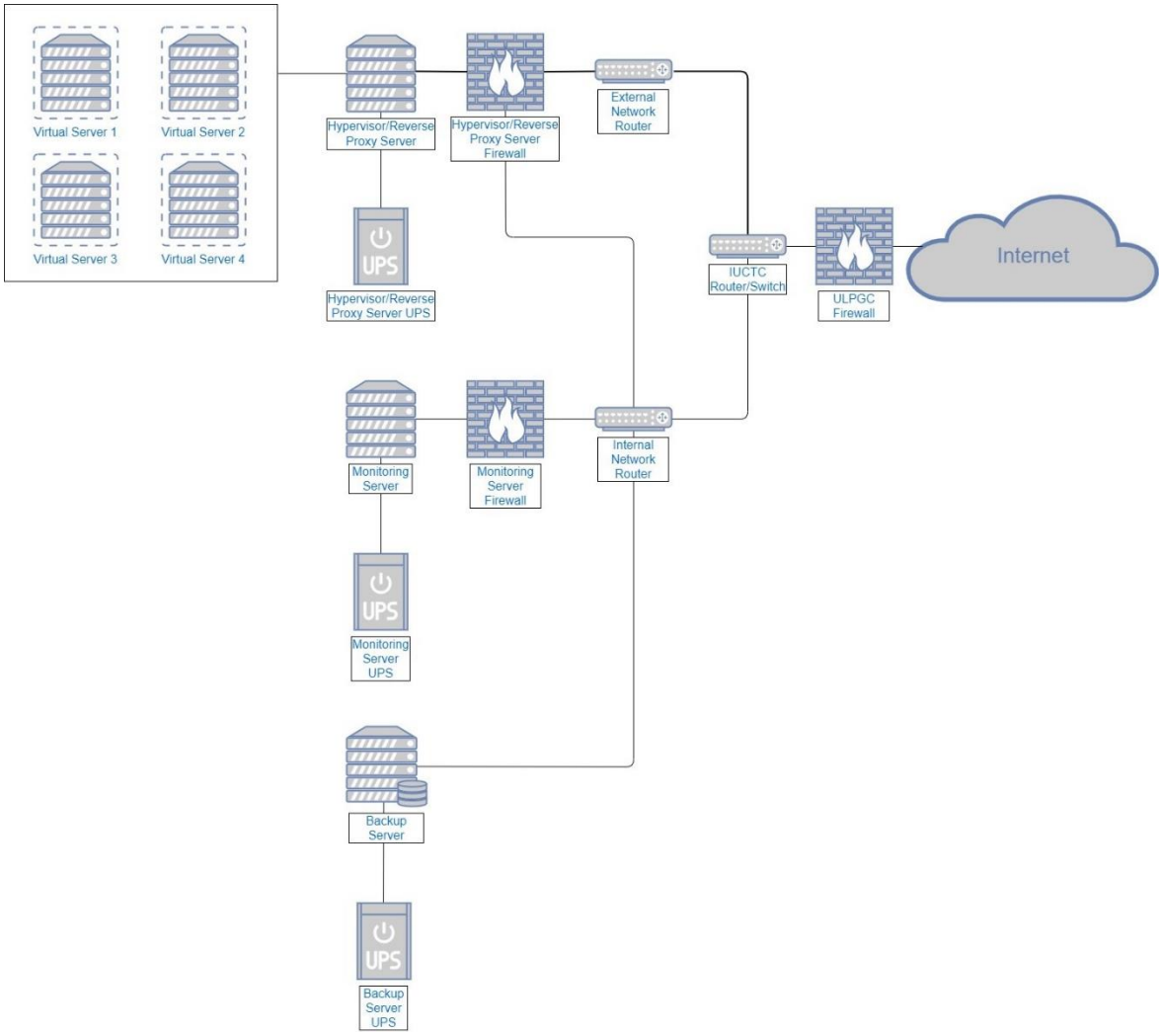


Ilustración 5: Esquema de red de la infraestructura diseñada para el proyecto

En el esquema dibujado en la ilustración se plantea la existencia de un servidor hipervisor (*Hypervisor/Reverse Proxy Server*) en el que se encontrarían todos los servidores web virtuales del centro, y el cual también tendría instalado un *proxy* inverso que permitiría a los clientes acceder a los servicios web alojados en dichos servidores virtuales.

Además, formaría parte de la infraestructura ideada un servidor de monitorización (*Monitoring Server*) que monitorizaría el estado del servidor hipervisor, sus recursos y los servicios web alojados en los servidores virtuales.

Cada uno de estos servidores físicos tendría integrado un cortafuegos que regularía las conexiones a los mismos.

Asimismo, se instalaría y configuraría un servidor de copias de seguridad (*Backup Server*) en el que luego se almacenarían las copias de seguridad de los datos de los servidores web virtuales.

Por último, se conectaría un sistema de alimentación ininterrumpida (*UPS*) a cada servidor físico de la nueva infraestructura.

9 CAPÍTULO 9: DESARROLLO

Como punto de partida, se decidió que el servidor hipervisor y *proxy* inverso del esquema de red planteado en este proyecto sería el servidor físico MR, debido a sus altas prestaciones, para lo cual se encontró necesario reinstalar su sistema operativo.

Sin embargo, esta acción requirió la previa instalación de un servidor web temporal que diera soporte a los servicios web alojados en el servidor MR mientras este estuviera fuera de servicio.

Tras haberse realizado la preparación del servidor web temporal, se migró y se desplegó en este los servicios web alojados en el servidor MR mientras se instalaba y configuraba en este último tanto el hipervisor de KVM como el *proxy* inverso de NGINX.

A continuación, se creó, instaló y configuró en el servidor MR cada uno de los servidores web virtuales en los que se alojarían los servicios web del centro.

Inmediatamente después se migró cada servicio web a su servidor virtual correspondiente, así como sus nombres de dominio, y se configuró el *proxy* inverso para posibilitar a los clientes el acceso a dichos servicios.

Por último, se implementó tanto el sistema de copias de seguridad de los datos de los servidores virtuales como el de apagado y encendido controlado de los servidores físicos y se instaló y configuró el servidor de monitorización de los servidores físicos y los servicios web de la nueva infraestructura del centro.

9.1 INSTALACIÓN DE UN SERVIDOR WEB TEMPORAL EN EL SERVIDOR NG

En primer lugar, se decidió que el servidor web temporal estaría instalado en el servidor físico NG.

El servidor NG fue conectado a la red pública del centro, pues no se estaba utilizando en ese momento para ningún fin.

El procedimiento seguido para la instalación de este servidor constó de cuatro fases: la realización de una copia de seguridad, la reinstalación del sistema operativo, la configuración del servidor y, por último, la migración a este de todos los servicios web desplegados en el servidor MR.

9.1.1 Copia de seguridad de los datos

El objetivo de esta copia de seguridad, en este caso, fue un conjunto de repositorios que estaban localizados en el directorio */var/repos/*.

Se decidió que esta copia de respaldo sería almacenada en el servidor de copias de seguridad B2. Para ello, primero fue necesario acceder a dicho servidor, donde luego se creó una carpeta compartida en la que, finalmente, se guardó la copia. Este procedimiento se encuentra detallado en el Anexo IV de este documento.

9.1.2 Reinstalación del sistema operativo

Una vez realizada la copia de seguridad de los datos del servidor NG, se reinstaló en este el sistema operativo CentOS 7 con el entorno gráfico GNOME.

Este procedimiento se encuentra detallado en el Anexo V de este documento.

9.1.3 Configuración

Para conformar el servidor NG como un servidor web, tras la reinstalación de su sistema operativo fue necesario acceder como usuario administrador a dicho servidor, en el cual luego se instaló y configuró la distribución de XAMPP. Este procedimiento se encuentra detallado en el Anexo VI de este documento.

9.1.4 Migración y despliegue de un servicio web

Durante esta fase se migraron todos y cada uno de los servicios web que estaban siendo alojados en ese preciso momento por el servidor MR al servidor NG.

Debido a que este proceso era bastante delicado y se pretendía mantener la disponibilidad el mayor tiempo posible, se decidió migrar los servicios web de uno en uno y solo durante el tiempo en el que no estaban siendo utilizados.

Finalmente, fue necesario desasociar los nombres de dominio de cada servicio migrado de la máquina MR y asociarlos a la máquina NG.

Todo este procedimiento se encuentra detallado en el Anexo VII de este documento.

9.2 INSTALACIÓN DE UN HIPERVISOR Y UN *PROXY* INVERSO EN EL SERVIDOR MR

Como se mencionó anteriormente, se decidió que el servidor hipervisor estaría instalado en el servidor físico MR.

El servidor MR tenía dos interfaces de red. Este estaba previamente conectado, mediante la interfaz de red 1, a la red pública del centro pues se estaba utilizando como servidor web en ese momento. Sin embargo, la interfaz de red 2 no estaba siendo utilizada, así que se conectó a la red interna del centro.

El procedimiento seguido para la instalación de este servidor constó de tres fases: la realización de una copia de seguridad, la reinstalación del sistema operativo y, por último, la configuración del servidor.

9.2.1 Copia de seguridad de los datos

El objetivo de esta copia de seguridad, en este caso, fue el conjunto de proyectos web que estaban localizados en el directorio */opt/lampp/htdocs/* del servidor MR.

Dicha copia de seguridad se guardó en un disco duro externo provisto por el instituto.

9.2.2 Reinstalación del sistema operativo

Una vez realizada la copia de seguridad de los datos del servidor MR, se reinstaló en este el sistema operativo CentOS 7 con el entorno gráfico GNOME.

Este procedimiento se encuentra detallado en el Anexo V de este documento.

9.2.3 Configuración

Para convertir el servidor MR tanto en un hipervisor como en un servidor de *proxy* inverso, tras la reinstalación de su sistema operativo fue necesario acceder como usuario administrador a dicho servidor, en el cual luego se instaló y configuró el hipervisor de KVM, así como el *proxy* inverso de NGINX.

Además, se instaló el servidor XRDP para facilitar la gestión de máquinas virtuales de forma remota desde cualquier equipo de la red interna del centro.

Este procedimiento se encuentra detallado en el Anexo VIII de este documento.

9.3 INSTALACIÓN DE LOS SERVIDORES WEB VIRTUALES

Tras la instalación y configuración del servidor hipervisor se procedió con la instalación de los servidores web virtuales.

Para la comunicación entre ellos y el hipervisor se utilizó una red, también virtual, de tipo NAT. De esta forma, los servidores virtuales se dispondrían en una red aislada con un rango de direcciones IP privadas, pero tendrían acceso a Internet mediante la dirección IP del servidor anfitrión o hipervisor.

Para la comunicación de los clientes con los servidores virtuales se utilizó el *proxy* inverso instalado en el servidor hipervisor.

El procedimiento seguido para la instalación de los servidores virtuales constó de cinco fases: la creación de un servidor virtual por cada servicio web del IUCTC, la instalación del sistema operativo de cada servidor virtual, la configuración de cada servidor virtual y, por último, la migración de cada servicio web del IUCTC a su propio servidor web virtual. Además, se actualizó la versión del servicio web de Moodle del centro.

9.3.1 Creación de los servidores virtuales

Para crear cada servidor virtual se accedió al escritorio gráfico de la máquina MR como usuario administrador y se siguió el procedimiento expuesto en el Anexo IX de este documento.

9.3.2 Instalación del sistema operativo de los servidores virtuales

Tras la creación de cada máquina virtual, se instaló en estas el sistema operativo CentOS7 en su versión mínima. Este procedimiento se hizo de manera similar al detallado en el Anexo V de este documento, con la única diferencia de que en el apartado "Selección de Software" se seleccionó "Instalación mínima".

9.3.3 Automatización del inicio de los servidores virtuales

Para automatizar el inicio de cada servidor virtual con el arranque del servidor MR se accedió a la interfaz de línea de comandos de la máquina MR como usuario administrador y se ejecutó, por cada servidor virtual creado, un comando con la siguiente estructura:

```
# virsh autostart --domain NOMBRE_SERVIDOR_VIRTUAL
```

9.3.4 Primera configuración del servidor virtual para el alojamiento del servicio web de *Moodle*: Instalación de Apache y MariaDB

Tras crear las máquinas virtuales correspondientes se procedió a realizar, en primer lugar, la primera configuración del servidor virtual que alojaría el servicio web de Moodle.

Dicha configuración consistió en la instalación de Apache y MariaDB para poder migrar el servicio web de Moodle del servidor original al servidor virtual.

Este procedimiento se encuentra detallado en el Anexo X de este documento.

9.3.5 Migración y despliegue del servicio web de *Moodle* en un servidor virtual

Durante esta fase se migró el servicio web de Moodle que estaba siendo alojado en ese preciso momento por el servidor MD.

Este proceso también requirió que el servicio no estuviera siendo utilizado.

Finalmente, fue necesario desasociar los dominios de acceso del servicio de Moodle del servidor MD y asociarlos al servidor MR.

Este procedimiento se encuentra detallado en el Anexo XI de este documento.

9.3.6 Actualización del servicio web Moodle a la versión 3.1

Durante esta fase se actualizó a la versión 3.1 la instancia de Moodle desplegada en el servidor virtual. Este procedimiento se encuentra detallado en el Anexo XII de este documento.

9.3.7 Segunda y última configuración del servidor virtual para el alojamiento del servicio web de *Moodle*: Instalación y configuración de XAMPP

Tras la la actualización de Moodle a la versión 3.1 se procedió a realizar la segunda y última configuración del servidor virtual de alojamiento del servicio web de Moodle.

Dicha configuración consistió en la instalación y configuración de XAMPP para poder actualizar el servicio web de Moodle del servidor virtual a la última de versión del momento bajo el servicio de XAMPP. Este procedimiento se encuentra detallado en el Anexo XIII de este documento.

9.3.8 Actualización del servicio web de Moodle a la versión 3.6

Durante esta fase se actualizó a la versión 3.6 la instancia de Moodle desplegada en el servidor virtual. Este procedimiento se encuentra detallado en el Anexo XIV de este documento.

9.3.9 Actualización del servicio web de Moodle a la versión 3.7

Durante esta fase se actualizó a la versión 3.7 la instancia de Moodle desplegada en el servidor virtual. Este procedimiento se encuentra detallado en el Anexo XV de este documento.

9.3.10 Configuración de los servidores virtuales para el alojamiento del resto de servicios web

Tras la actualización del servicio Moodle a la versión más reciente se procedió a configurar el resto de los servidores virtuales que alojarían cada uno de los servicios web restantes.

Para conformar cada servidor virtual como un servidor web, fue necesario acceder como usuario administrador a dicho servidor, en el cual luego se instaló y configuró la distribución de XAMPP. Este procedimiento se encuentra detallado en el Anexo XVI de este documento.

9.3.11 Migración y despliegue de un servicio web en un servidor virtual

Para migrar un servicio web antes fue necesario asegurarse de que este fuera inaccesible.

Cada servicio se migró de uno en uno y solo durante el tiempo en el que no estaban siendo utilizados.

Finalmente, fue necesario desasociar los dominios de acceso de cada servicio migrado de los servidores originales y asociarlos al servidor MR.

Este procedimiento se encuentra detallado en el Anexo XVII de este documento.

9.4 IMPLEMENTACIÓN DEL SISTEMA DE COPIAS DE SEGURIDAD DE LOS SERVIDORES WEB VIRTUALES EN EL SERVIDOR MR

Antes que nada, se decidió que el lugar de almacenamiento temporal de las copias de seguridad sería el servidor de copias de seguridad B2.

Debido a que no se encontró software libre adecuado para la realización de las copias de seguridad de los servidores web virtuales de la nueva infraestructura fue necesario desarrollar, por cuenta propia, un sistema de copias de seguridad específico para el caso el cual operaría desde el servidor MR.

El procedimiento seguido para la implementación del sistema constó de tres fases: la creación de una carpeta compartida en el servidor B2, el desarrollo del sistema de copias de seguridad de los servidores web virtuales y, por último, el inicio del sistema de copias de seguridad de los servidores web virtuales.

9.4.1 Creación de la carpeta compartida *CopiasMV* en el servidor B2

Se creó una carpeta compartida llamada "CopiasMV" en el servidor de copias de seguridad B2 que serviría para almacenar todas las copias de seguridad generadas por el sistema de copias de seguridad.

Este procedimiento se encuentra detallado en el Anexo XVIII de este documento.

9.4.2 Desarrollo del sistema de copias de seguridad

Como se comentó anteriormente, fue necesario desarrollar un sistema de copias de seguridad particular que se encargara de respaldar los datos de cada servidor web virtual desplegado en la nueva infraestructura. Para ello se tomó la iniciativa de crear varios ficheros de órdenes o *scripts* en el servidor MR que, en su conjunto, cumplieran esta función correctamente.

Los ficheros creados fueron los siguientes:

- El *script* principal de copias de seguridad: *vmbackup.sh*. Este se encargaría de realizar todo el proceso de copiar el fichero XML y las imágenes de disco de cada servidor virtual al servidor de copias de seguridad B2.
- El *script* de restauración de copias de seguridad: *vmrestore.sh*. Este se encargaría de restaurar una copia de seguridad determinada de un servidor virtual desde el servidor de copias de seguridad B2 al servidor MR.
- El *script* de limpieza de copias de seguridad: *vmbackup_clean.sh*. Este se encargaría de limpiar las copias de seguridad más antiguas del servidor B2 liberando así espacio en el disco para el almacenamiento de nuevas copias.
- El fichero de registro o *log* de todas las acciones llevadas a cabo por el sistema de copias de seguridad: *vmbackup.log*. Este se encargaría de almacenar de forma cronológica todos los eventos producidos por los *scripts* conformados en el sistema de copias de seguridad.

Cada uno de los *scripts* fue desarrollado utilizando el lenguaje de consola de Bash.

Este procedimiento se encuentra detallado en el Anexo XIX de este documento.

9.4.3 Inicio del sistema de copias de seguridad

Por último, fue necesario programar la ejecución tanto del *script vmbackup.sh*, para realizar copias de seguridad periódicamente en el servidor de copias de seguridad B2, como del *script vmbackup_clean.sh*, para realizar una limpieza periódica de las copias de

seguridad más antiguas de dicho servidor y evitar el agotamiento de espacio disponible para nuevas copias.

Este procedimiento se encuentra detallado en el Anexo XX de este documento.

9.5 INSTALACIÓN DEL SISTEMA DE MONITORIZACIÓN EN EL SERVIDOR MD

En primer lugar, se decidió que el sistema de monitorización estaría instalado en el servidor físico MD. Este ya estaba previamente conectado a la red pública del centro, pues se estaba utilizando como servidor web en ese momento.

Este servidor monitorizaría, en un primer momento, el estado de servicio del servidor MR, así como el estado de algunos servicios web del centro.

El procedimiento seguido para la instalación de este servidor constó de tres fases: la realización de una copia de seguridad, la reinstalación del sistema operativo y, por último, la configuración del servidor.

9.5.1 Copia de seguridad de los datos

El objetivo de esta copia de seguridad, en este caso, fue el conjunto de proyectos web que estaban localizados en el directorio */opt/lampp/htdocs/* del servidor MD.

Dicha copia de seguridad se guardó en un disco duro externo provisto por el instituto.

9.5.2 Reinstalación del sistema operativo

Una vez realizada la copia de seguridad de los datos del servidor MD, se reinstaló en este el sistema operativo CentOS 7 con el entorno gráfico GNOME.

Este procedimiento se encuentra detallado en el Anexo V de este documento.

9.5.3 Configuración

Para conformar el servidor MD como un servidor de monitorización, tras la reinstalación de su sistema operativo fue necesario acceder como usuario administrador a dicho servidor, en el cual luego se instaló y configuró el sistema de monitorización de Zabbix, con cierta ayuda de una guía publicada por DigitalOcean [40].

Este procedimiento se encuentra detallado en el Anexo XXI de este documento.

9.6 IMPLEMENTACIÓN DEL SISTEMA DE APAGADO Y ENCENDIDO CONTROLADO DE LOS SERVIDORES FÍSICOS NG, MR Y MD

Tras la finalizar la instalación de cada uno de los servidores de la nueva infraestructura se procedió a conectar cada servidor físico a un sistema de alimentación ininterrumpida (SAI o UPS en adelante) individual. Cada uno de los servidores físicos NG, MR y MD fue conectado a su respectivo UPS mediante la fuente de alimentación y un cable USB.

El procedimiento seguido para la implementación de estos sistemas constó de dos fases: la instalación del software de gestión de los UPS y la configuración y programación de dicho software para el encendido y apagado controlado de los servidores físicos utilizados en la nueva infraestructura.

9.6.1 Instalación de PowerShield^3

En primer lugar, fue necesario instalar en cada uno de los servidores físicos el software de gestión de su respectivo UPS.

El proceso de instalación de PowerShield^3 se encuentra detallado en el Anexo XXII de este documento.

9.6.2 Configuración de PowerShield^3

Seguidamente, se configuró PowerShield^3 de forma que cada UPS realizara un apagado controlado del sistema respaldado 20 minutos despúes de que el UPS fuera desconectado de la corriente o 5 minutos antes de que la batería del UPS se fuera a agotar mientras que el propio UPS se apagaría 5 segundos después del apagado completo del sistema respaldado.

Este procedimiento de configuración se encuentra detallado en el Anexo XXIII de este documento.

9.6.3 Inicio de PowerShield^3

A continuación, se inició el software de gestión de los UPS en los servidores NG, MR y MD:

```
# systemctl start upsmon
```

9.6.4 Automatización del inicio de PowerShield^3

Por último, se automatizó el inicio el software de gestión de los UPS en los servidores NG, MR y MD:

```
# systemctl enable upsmon
```

10 CAPÍTULO 10: EVALUACIÓN

Tras completar la implementación de la infraestructura basada en virtualización ideada para este proyecto se procedió a realizar una evaluación de los sistemas integrados en la misma.

El procedimiento seguido para la evaluación de los sistemas de la nueva infraestructura constó de cuatro fases: la evaluación del rendimiento de los servidores físicos y la disponibilidad de los servicios web, la evaluación del funcionamiento del sistema de copias de seguridad, la evaluación del funcionamiento del sistema de monitorización y, por último, la evaluación del funcionamiento del sistema de encendido y apagado controlado de los servidores físicos.

10.1 EVALUACIÓN DEL RENDIMIENTO DE LOS SERVIDORES FÍSICOS Y LA DISPONIBILIDAD DE LOS SERVICIOS WEB DE LA NUEVA INFRAESTRUCTURA

Para analizar el funcionamiento de la infraestructura implementada se decidió realizar dos comprobaciones: una sobre el rendimiento de los servidores físicos que la componían y otra sobre la disponibilidad de los servicios web que ofrecía.

10.1.1 Evaluación del rendimiento de los servidores físicos

Para evaluar el rendimiento de los servidores de la nueva infraestructura solo se necesitó realizar un estudio del índice de utilización de recursos del servidor MR, pues era el servidor principal de la nueva infraestructura en el que se alojaban todos los servidores web virtuales del centro, después de que los servidores físicos MC y WN fueran apagados.

Para ello, se desarrolló y se programó de la misma manera la ejecución del *script,* denominado *top*.sh, utilizado anteriormente en la fase de análisis, para monitorizar la utilización de los recursos del servidor a lo largo del tiempo.

Los resultados de este estudio se utilizaron para comparar, esta vez, el índice de utilización del servidor MR en la infraestructura original con el índice de utilización del mismo servidor operando en la nueva infraestructura. Para ello, se realizaron una serie de gráficos con los resultados obtenidos del fichero */usr/local/bin/SoftMon/top.txt* durante el último mes de desarrollo del proyecto, se hizo un promedio del porcentaje de uso de los recursos del servidor MR durante cada día y fueron comparados con los obtenidos del servidor MR de la infraestructura original.

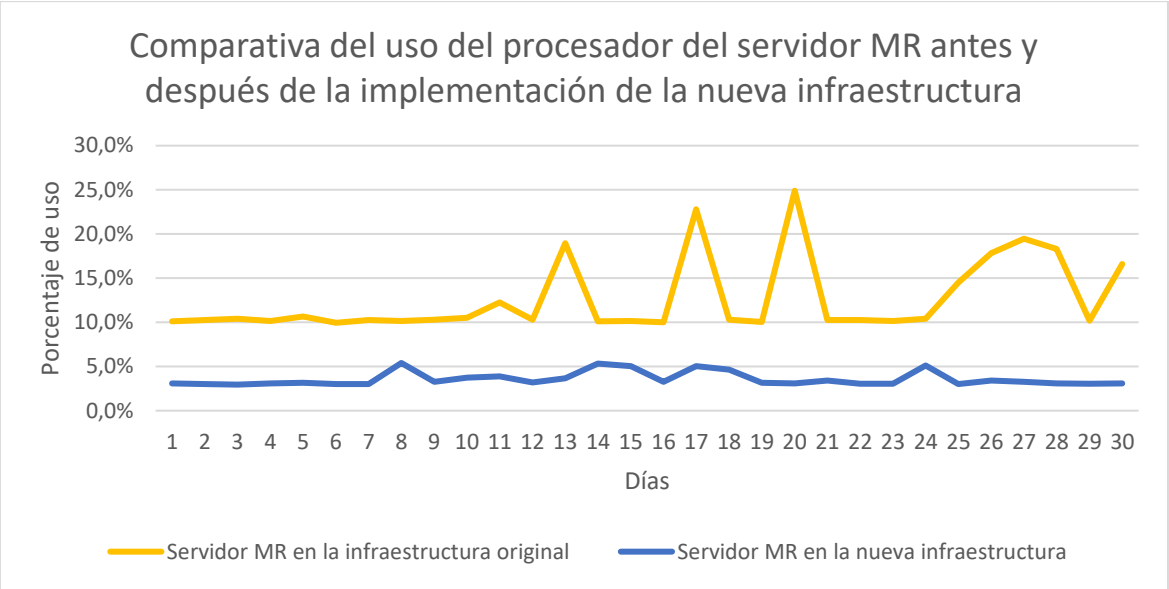


Ilustración 6: Gráfico del uso de procesador del servidor MR antes y despúes de la implementación de la nueva infraestructura

El gráfico anterior (Ilustración 6) muestra un mayor índice de utilización del procesador del servidor MR en la infraestructura original.

Esta diferencia se debe a que en la infraestructura original el servidor ejecutaba procesos, correspondientes a los servicios web que se alojaban, de forma activa, mientras que en la nueva infraestructura su porcentaje de uso se vio reducido a causa de la gran cantidad de procesos, correspondientes a los servidores virtuales, que estaban en estado de espera (*interruptible sleep*) en cada momento.

Este aumento en el número de procesos en estado de espera fue consecuencia de la propiedad de la virtualización que permite a las máquinas virtuales compartir de una manera bastante eficiente el uso de los procesadores de la máquina anfitriona.

De esta forma se pudo interpretar la disminución del uso de los procesadores del servidor en la nueva infraestructura como un ajuste a la demanda de procesamiento de los servidores virtuales.

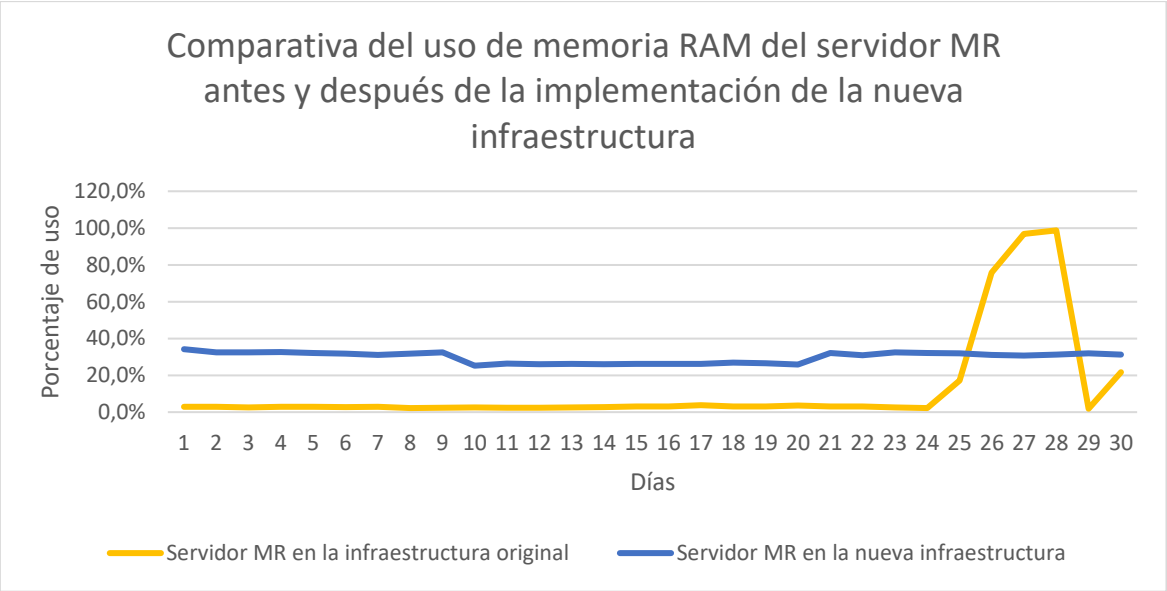


Ilustración 7: Gráfico del uso de memoria RAM del servidor MR antes y despúes de la implementación de la nueva infraestructura

El gráfico anterior (Ilustración 7) muestra un índice de utilización de la memoria RAM del servidor MR notablemente mayor en la nueva infraestructura como resultado de la demanda de memoria de los servidores virtuales en ejecución en dicho servidor.

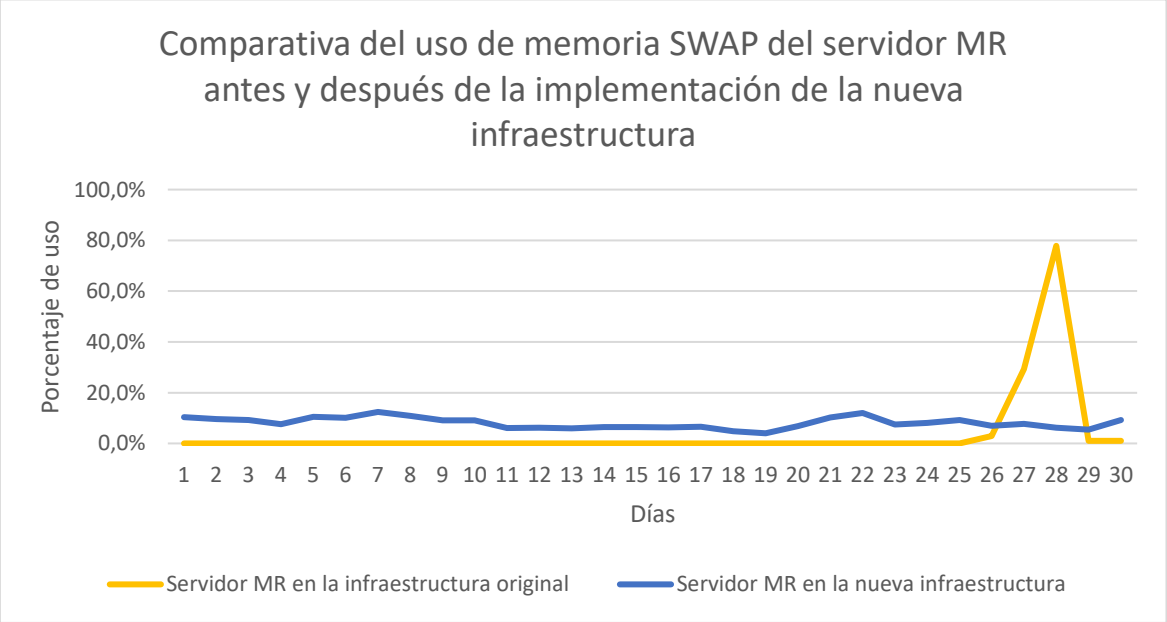


Ilustración 8: Gráfico del uso de memoria RAM del servidor MR antes y despúes de la implementación de la nueva infraestructura

El gráfico anterior (Ilustración 8) muestra un índice de utilización de la memoria SWAP del servidor MR moderadamente mayor en la nueva infraestructura como resultado de la demanda de memoria de los servidores virtuales en ejecución en dicho servidor.

Este aumento se debió a la característica de *overcommitting* de memoria RAM de KVM que permitía a los procesos de las máquinas virtuales utilizar la memoria SWAP en caso de que necesitara más memoria para su ejecución o, por el contrario, en caso de que dichos procesos se encontraran en estado de espera y no se requiriera que estuvieran cargados en la memoria RAM.

Por lo tanto, se pudo establecer una relación entre los numerosos procesos, correspondientes a los servidores virtuales, en estado de espera del servidor en la nueva infraestructura y el aumento del uso de la memoria SWAP del servidor, ya que era en esta última donde se guardaban dichos procesos.

10.1.2 Evaluación de la disponibilidad de los servicios web

Para comprobar la disponibilidad de los servicios web simplemente se accedió mediante un navegador web a cada uno de ellos utilizando su nombre de dominio asociado y comprobando que respondiera correctamente en el tiempo esperado.

10.2 EVALUACIÓN DEL FUNCIONAMIENTO DEL SISTEMA DE COPIAS DE SEGURIDAD DE LOS SERVIDORES WEB VIRTUALES IMPLEMENTADO EN LA NUEVA INFRAESTRUCTURA

Para evaluar el funcionamiento del sistema de copias de seguridad de los servidores web virtuales implementado en la nueva infraestructura se decidió realizar cuatro comprobaciones de funcionamiento, una por cada fichero del sistema.

Debido a que este sistema se encontraba integrado bajo el directorio */usr/local/bin/vmbackuptools/* del servidor MR fue necesario acceder a este como usuario administrador para realizar cada comprobación.

Además, fue necesario crear una máquina virtual adicional en el servidor MR llamada *NOMBRE_MV* que sería utilizada para la realización de varias pruebas.

10.2.1 Evaluación del funcionamiento del *script* principal de copias de seguridad: vmbackup.sh

Se comprobó que el *script* principal de copias de seguridad *vmbackup.sh* del sistema de copias de seguridad funcionara correctamente.

Estas comprobaciones se exponen en el Anexo XXIV de este documento.

10.2.2 Evaluación del funcionamiento del *script* de limpieza de copias de seguridad: vmbackup_clean.sh

Se comprobó que el *script* de limpieza de copias de seguridad *vmbackup_clean.sh* del sistema de copias de seguridad funcionara correctamente.

Estas comprobaciones se exponen en el Anexo XXV de este documento.

10.2.3 Evaluación del funcionamiento del *script* de restauración de copias de seguridad: vmrestore.sh

Se comprobó que el *script* interactivo de restauración de copias de seguridad *vmrestore.sh* del sistema de copias de seguridad funcionara correctamente.

Estas comprobaciones se exponen en el Anexo XXVI de este documento.

Cabe destacar que tras la comprobación del funcionamiento de cada *script* se comprobó el correcto funcionamiento de cada uno de los servidores web virtuales, así como la disponibilidad de los servicios web que alojaban.

10.3 EVALUACIÓN DEL FUNCIONAMIENTO DEL SISTEMA DE MONITORIZACIÓN IMPLEMENTADO EN LA NUEVA INFRAESTRUCTURA

Para evaluar el funcionamiento del sistema de monitorización implementado en la nueva infraestructura se decidió realizar dos comprobaciones: una sobre el funcionamiento de la monitorización de los equipos y otra sobre el funcionamiento de la monitorización de los servicios web.

10.3.1 Evaluación del funcionamiento de la monitorización de equipos

Para evaluar el estado de funcionamiento de la característica de monitorización de equipos de Zabbix se realizaron una serie de comprobaciones que se exponen en el Anexo XXVII de este documento.

10.3.2 Evaluación del funcionamiento de la monitorización de servicios web

Para evaluar el estado de funcionamiento de la característica de monitorización de servicios web de Zabbix se realizaron una serie de comprobaciones que se exponen en el Anexo XXVIII de este documento.

10.4 EVALUACIÓN DEL FUNCIONAMIENTO DEL SISTEMA DE APAGADO Y ENCENDIDO CONTROLADO DE LOS SERVIDORES FÍSICOS IMPLEMENTADO EN LA NUEVA INFRAESTRUCTURA

Para evaluar el funcionamiento del sistema de de apagado y encendido controlado implementado en la nueva infraestructura se decidió realizar una prueba consistente en desconectar de la corriente uno de los sistemas de alimentación ininterrumpida conectados a los servidores físicos de la misma.

De esta forma se comprobó que:

- El SAI proporcionara energía durante 20 minutos a su respectivo servidor físico tras la desconexión del sistema de la corriente.
- El SAI apagara correctamente su respectivo servidor físico transcurridos 20 minutos desde la desconexión del sistema de la corriente.
- El SAI se apagara 5 segundos después del apagado total de su respectivo servidor físico tras los 20 minutos iniciales de desconexión del sistema de la corriente.

Tras realizar estas comprobaciones se volvió a conectar el SAI a la corriente y se verificó que, inmediatamente después de la restauración de la corriente, el servidor físico conectado a este volviera a encenderse de forma automática.

11 CAPÍTULO 11: CONCLUSIONES Y TRABAJOS FUTUROS

11.1 CONCLUSIONES

Este proyecto nació de la necesidad de optimizar la infraestructura del Instituto Universitario de Ciencias Tecnológicas y Cibernéticas, para lo cual se establecieron varios objetivos cuya consecución llevaron su realización a buen término.

¿Cómo se alcanzaron los objetivos propuestos?

En primer lugar, se realizó un estudio minucioso de los sistemas y servicios de la infraestructura original, así como los recursos a disposición del instituto.

Con la información recabada en dicho estudio, se realizó un diseño viable de la infraestructura basada en virtualización mediante un esquema de red en el que se incluyeron todos sistemas principales que conformarían la misma y se estableció claramente la función de cada uno.

Para implementar dicha infraestructura fue necesario la utilización de un servidor como servidor principal en el que se instaló un hipervisor, en este caso de KVM, a partir del cual se crearon y configuraron las máquinas virtuales que actuarían como los nuevos servidores web del centro. Esto requirió la migración controlada de los servicios web del centro desde los servidores originales a los virtuales, así como sus correspondientes nombres de dominio.

Adicionalmente, se requirió la instalación en el servidor principal de un *proxy* inverso, en este caso de NGINX, que permitiera a los clientes poder acceder a los servicios alojados en los servidores web virtuales de la nueva infraestructura.

Para la implantación de un sistema de copias de seguridad que respaldara los datos de los servidores virtuales se desarrollaron y se programó la ejecución de varios *scripts* mediante Bash en el servidor principal que se encargaran de realizar copias diarias de los ficheros de configuración y las imágenes de disco de los servidores virtuales para luego almacenarlas en uno de los servidores de copias de seguridad del centro.

Para la implantación de un sistema de monitorización se utilizó un servidor físico adicional en el que se instaló el sistema de monitorización de Zabbix. Además, se instaló el agente de monitorización de Zabbix en cada servidor monitorizado. Con ello y con las configuraciones pertinentes realizadas se pudo comenzar a monitorizar el servidor principal y varios de los servicios web del centro.

Para la implantación de los sistemas de encendido y apagado controlado simplemente se conectó cada servidor de la nueva infraestructura a un SAI independiente de los muchos de los que disponía el centro y se configuraron instalando en dichos servidores el software de gestión correspondiente.

Por último, se realizó una serie de evaluaciones que reflejaron, además del correcto funcionamiento de cada uno de los sistemas de la nueva infraestructura, la diferencia en el índice de utilización del servidor principal antes y después de la implementación de la infraestructura basada en virtualización y revelando un mejor aprovechamiento de los recursos del mismo a partir del momento en el que se comenzó a utilizar como hipervisor y *proxy* inverso.

Con dichas evaluaciones se pudo observar una optimización tanto en el índice de utilización como en el índice de tolerancia a fallos de la infraestructura del IUCTC con respecto a la original, además de una reducción en los costes de mantenimiento de los sistemas del centro al utilizar más servidores virtuales que servidores físicos. A esto se le suman todas las ventajas de la utilización de un soporte virtual.

A modo de conclusión, podemos decir que se ha realizado este proyecto de forma bastante satisfactoria al haber podido cumplir los objetivos propuestos resolviendo todos los problemas que han ido surgiendo a lo largo del desarrollo de este.

11.2 TRABAJOS FUTUROS

La implementación de nuevas tecnologías en la infraestructura del Instituto Universitario de Ciencias y Tecnologías Cibernéticas que ha supuesto la realización de este proyecto ha traído consigo una infinidad de opciones de configuración de los sistemas del centro que no existían en la infraestructura original.

Esto se debe a que varios de los sistemas que constituyen la infraestructura implementada en este proyecto tienen la potencia para realizar muchas más funciones que las contempladas en este documento. Se proponen a continuación varios ejemplos de posibles mejoras que podrían llevarse a cabo en dichos sistemas:

- Configuración del *proxy* inverso del servidor MR para asegurar el tráfico de HTTP de todos los servidores web virtuales (HTTPS) mediante un solo certificado digital.
- Configuración del *proxy* inverso del servidor MR para mitigar ataques DDoS a los servidores web virtuales.

- Configuración del *proxy* inverso del servidor MR para balancear la carga de un clúster de servidores web virtuales.
- Configuración del sistema de monitorización Zabbix del servidor MD para monitorizar una gran cantidad de equipos y servidores, recursos y aplicaciones web.
- Configuración del sistema de monitorización Zabbix del servidor MD para notificar infinidad de eventos y realizar gráficos personalizados con los datos obtenidos de los equipos monitorizados.
- Configuración del sistema de monitorización Zabbix del servidor MD para realizar mapas de red de los equipos monitorizados.

ANEXOS

ANEXO I: COMPROBACIÓN DE LAS ESPECIFICACIONES BÁSICAS DE LAS MÁQUINAS DE LA INFRAESTRUCTURA ORIGINAL

Comprobación del sistema operativo de las máquinas con distribuciones Ubuntu

En el caso de aquellas máquinas con distribución Ubuntu, se obtuvo la información del sistema operativo de la siguiente manera:

```
$ sudo lsb_release -a
```

Comprobación del sistema operativo de las máquinas con distribuciones CentOS

En el caso de aquellas máquinas con distribución CentOS, se obtuvo la información del sistema operativo de la siguiente manera:

```
# cat /etc/centos-release
```

Comprobación del nombre de la máquina

A continuación, se comprobó el nombre de cada máquina:

```
# cat /etc/hostname
```

Comprobación del número de procesadores

Posteriormente, se comprobó el número de procesadores de cada máquina, reales o virtuales, y otros datos asociados a estos:

```
# cat /proc/cpuinfo
```

Comprobación de la capacidad de almacenamiento en disco

Además, se comprobó la capacidad de almacenamiento de cada máquina, así como el espacio disponible y en uso del disco duro, entre otros datos:

```
# df -h
```

Comprobación de la capacidad de memoria

Por último, se comprobó la capacidad de memoria de cada máquina, entre otros datos:

```
# demidecode --type 17
```

Adicionalmente se obtuvieron y contrastaron otros datos de carácter más crítico referentes a los fabricantes y modelos de las máquinas, usuarios y contraseñas, etc., que se han decidido mantener, por razones de seguridad, en la confidencialidad del instituto.

ANEXO II: COMPROBACIÓN DEL ESTADO DE LOS SERVICIOS WEB DE LA INFRAESTRUCTURA ORIGINAL

Comprobación de servicios web de los servidores con distribuciones Ubuntu

Se verificó que todos aquellos servidores con distribución Ubuntu como sistema operativo utilizaban Apache como servidor web y MySQL como sistema de bases de datos.

En primer lugar, se comprobó si el servicio de servidor web, Apache, estaba en funcionamiento:

```
$ sudo service apache2 status
```

A continuación, se comprobó si el servicio de sistema de gestión de bases de datos, MySQL, estaba en funcionamiento:

```
$ sudo service mysql status
```

Por último, se hizo un listado del directorio */var/www/*, donde se encontraban los directorios de los documentos HTML relativos a los servicios web desplegados en cada servidor:

```
$ sudo ls /var/www/
```

Comprobación de servicios web de los servidores con distribuciones CentOS

Se verificó que todos aquellos servidores con distribución CentOS como sistema operativo utilizaban XAMPP como servidor web y sistema de bases de datos.

En primer lugar, se comprobó si el conjunto de servicios de XAMPP (Apache, MySQL y ProFTPD) estaba en funcionamiento:

```
# /opt/lampp/lampp status
```

Por último, se hizo un listado del directorio */opt/lampp/htdocs/*, donde se encontraban los directorios de los documentos HTML relativos a los servicios web desplegados en cada servidor:

```
# ls /opt/lampp/htdocs
```

ANEXO III: CREACIÓN Y PROGRAMACIÓN DE LA EJECUCIÓN DEL *SCRIPT* DE MONITORIZACIÓN DE RECURSOS *TOP.SH*

Creación y desarrollo del script de monitorización básica top.sh

En primer lugar, se creó el directorio */usr/local/bin/SoftMon/* en cada uno de los servidores:

```
# mkdir /usr/local/bin/SoftMon
```

Se creó en dicho directorio un fichero de texto llamado *top.txt*:

```
# touch /usr/local/bin/SoftMon/top.txt
```

Además, en el mismo directorio, se creó otro fichero llamado *top.sh*:

```
# touch /usr/local/bin/SoftMon/top.sh
```

Finalmente, se le otorgó a este último el permiso de ejecución:

```
# chmod +x /usr/local/bin/SoftMon/top.sh
```

En dicho fichero se desarrolló un pequeño *script* que se encargaba, haciendo uso del comando *top* de Linux, de obtener información de la utilización de recursos del sistema (CPU, memoria, procesos, …) del servidor correspondiente y guardarla en el fichero de texto */usr/local/bin/SoftMon/top.txt*.

El código fuente de dicho *script* se expone a continuación:

```
#!/bin/bash

# Se imprime la fecha actual en el fichero correspondiente
printf "$(date +"%d-%m-%Y")\n\n" >> /root/SoftMon/top.txt

# Se imprimen las 17 primeras líneas (con información relevante, excepto la cabecera) del comando top en modo "batch" en el fichero correspondiente
top -b -n1 | head -17 >> /root/SoftMon/top.txt

# Se imprime un separador en el fichero correspondiente
echo -e $x{0..45}"_" >> /root/SoftMon/top.txt
```

Programación de la ejecución del script top.sh mediante Cron

Mediante el administrador de procesos de Cron se programó para que se ejecutara dicho *script* cada hora en cada uno de los servidores. Para ello se modificó el fichero *crontab* de cada servidor:

```
# crontab -e
```

Y se añadió la siguiente línea:

```
0 * * * * /usr/local/bin/SoftMon/top.sh
```

ANEXO IV: COPIA DE SEGURIDAD DE LOS DATOS DEL DIRECTORIO */VAR/REPOS/* DEL SERVIDOR NG DE LA INFRAESTRUCTURA ORIGINAL EN EL SERVIDOR DE COPIAS DE SEGURIDAD B2

Acceso web al servidor B2

Para acceder al servidor B2, previamente hubo que conectarlo a la red privada del centro, de forma que solo pudiera ser accesible desde dicha red.

Una vez encendido, se accedió a la página de inicio de la interfaz web del servidor (Ilustración 9).

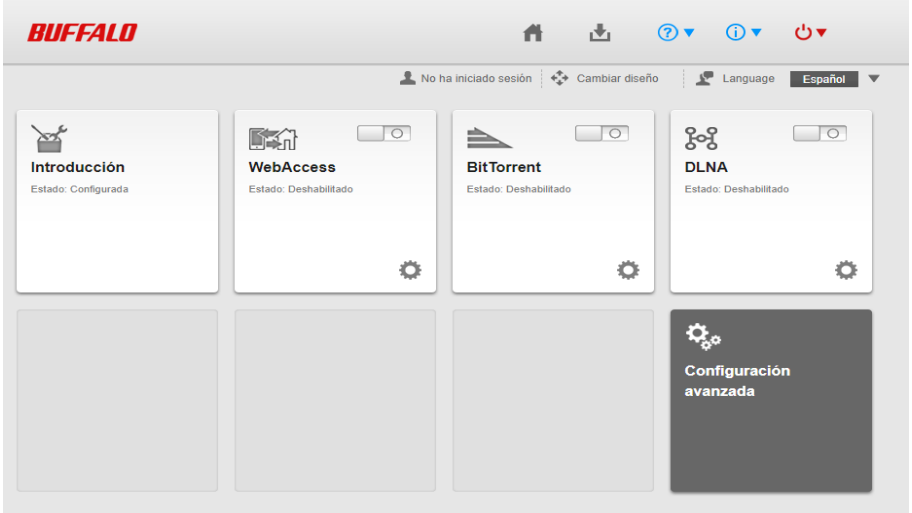


Ilustración 9: Página de inicio de la interfaz web del servidor de copias de seguridad B2

Creación de la carpeta compartida ReposSubv en el servidor B2

Para crear una carpeta compartida en el servidor B2 fue necesario, mediante unas credenciales válidas, acceder al apartado "Configuración avanzada" (Ilustración 10).

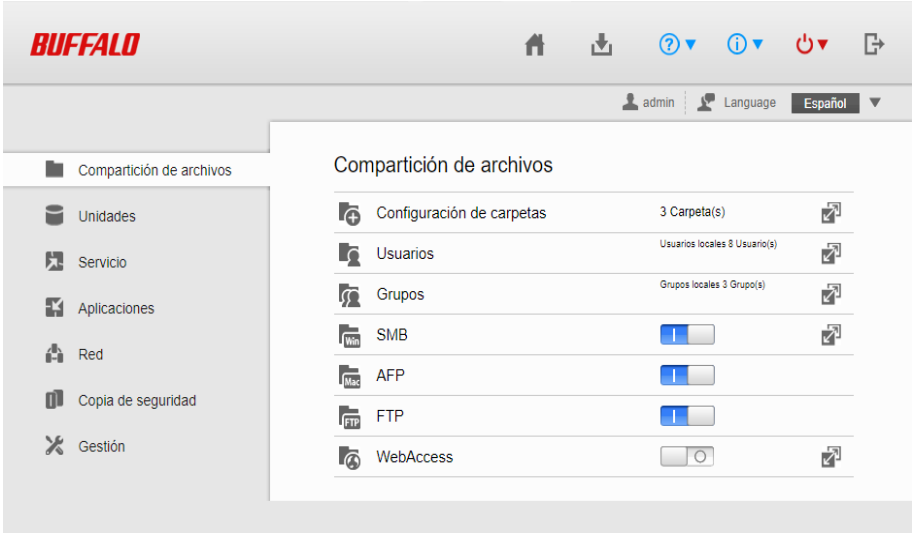


Ilustración 10: Página de configuración avanzada del servidor de copias de seguridad B2

En esta página se clicó en el botón de "Configuración de carpetas" y se accedió a la lista de carpetas compartidas del servidor B2, desde donde se creó una carpeta compartida llamada *ReposSubv*, permitiéndosele las conexiones mediante el protocolo Samba y FTP para poder transferir a la misma los ficheros de la copia de seguridad (Ilustración 11).

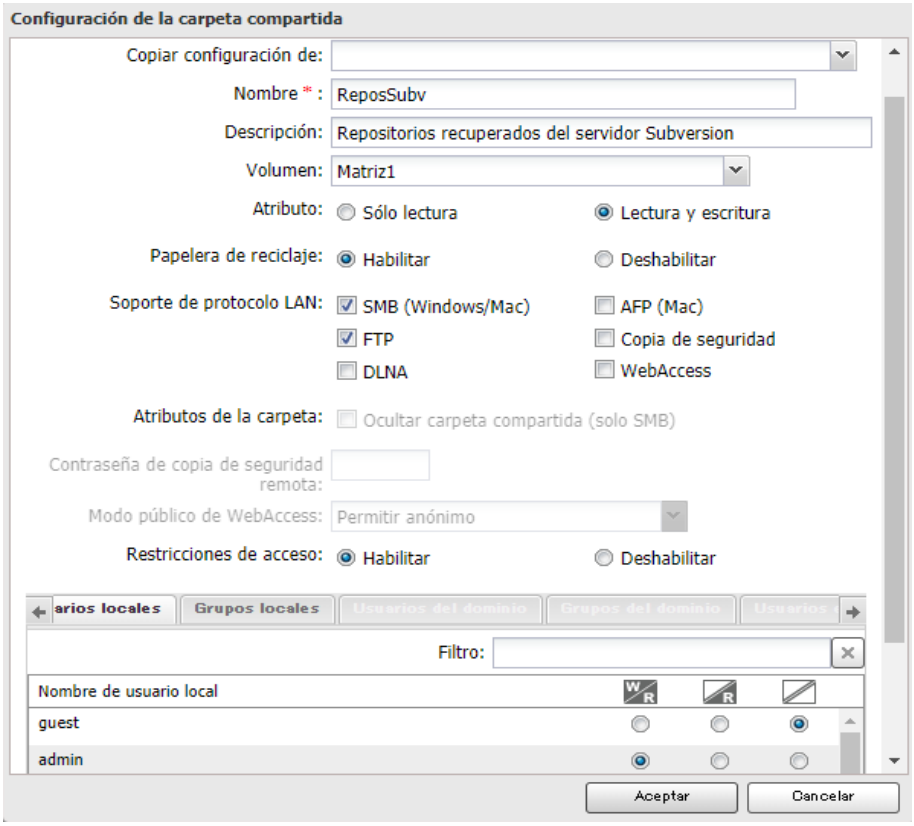


Ilustración 11: Página de creación de la carpeta compartida "ReposSubv" en el servidor de copias de seguridad B2

Instalación de LFTP

Para la transferencia de la copia de seguridad al servidor B2 se optó, como se mencionó antes, por utilizar el protocolo FTP. Con vistas a la realización de una transferencia de archivos de manera segura se decidió instalar la herramienta LFTP en el servidor NG.

Para ello, se accedió a dicho servidor como usuario *root* y, en la interfaz de línea de comandos, se ejecutó el siguiente comando:

```
# yum install lftp -y
```

Acceso FTP al servidor B2

A continuación, se utilizó el programa LFTP en el servidor NG para acceder mediante el protocolo FTP al servidor B2:

```
# lftp ftp://NOMBRE_USUARIO@DIRECCIÓN_IP_SERVIDOR_B2
```

Transferencia de archivos al servidor B2

Una vez dentro del sistema de ficheros del servidor B2, se utilizó una herramienta incorporada en el programa LFTP, denominada *mirror*, para transferir el directorio */var/repos* y su contenido desde el servidor NG a la carpeta compartida *ReposSubv* del servidor B2:

```
lftp NOMBRE_USUARIO@DIRECCIÓN_IP_SERVIDOR_B2:-> mirror -R /var/repos/
/array1/ReposSubv/
```

Al acabar la transferencia se finalizó la conexión FTP con el servidor B2:

```
lftp NOMBRE_USUARIO@DIRECCIÓN_IP_SERVIDOR_B2:-> exit
```

ANEXO V: INSTALACIÓN DEL SISTEMA OPERATIVO CENTOS 7 CON EL ENTORNO GRÁFICO GNOME

En primer lugar, justo al comenzar la instalación, se seleccionó el idioma utilizado en el proceso de instalación (Ilustración 12).

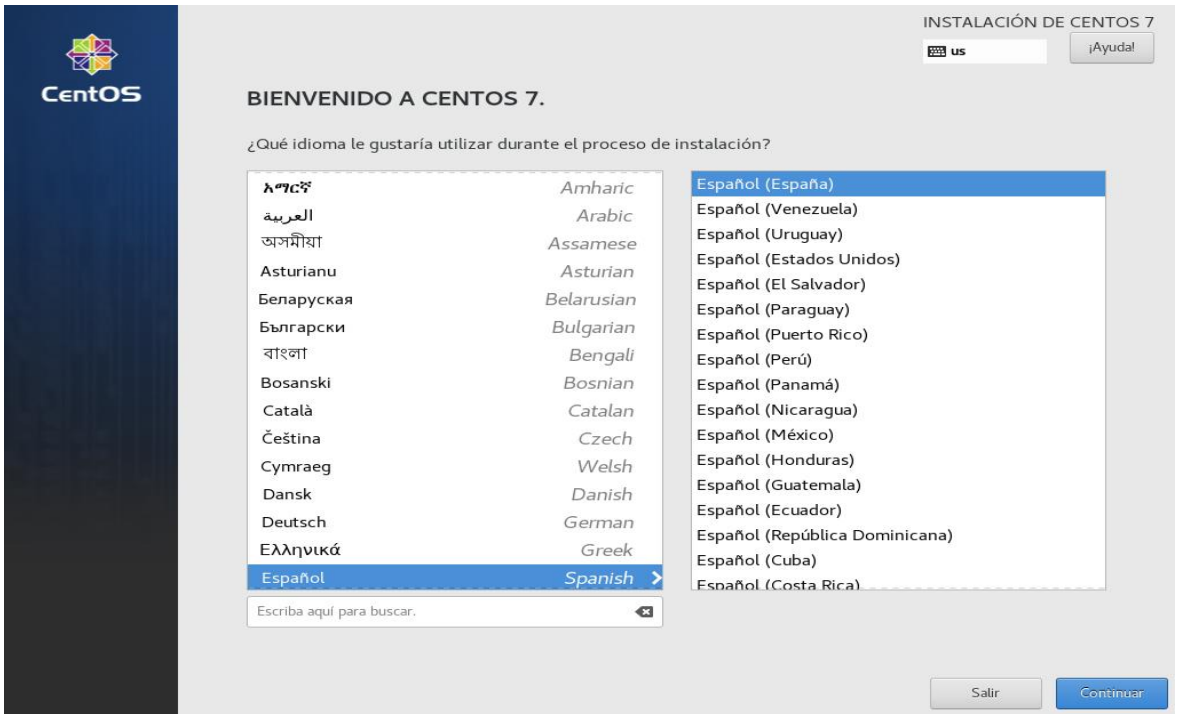


Ilustración 12: Proceso de instalación de CentOS 7 - Elección de idioma

Tras ello, se mostró la pantalla de configuración de la instalación (Ilustración 13).

Ilustración 13: Proceso de instalación de CentOS 7 - Resumen de configuración

A continuación, se configuró el apartado “Red y Nombre de Equipo” para habilitar la conexión por cable o Ethernet (Ilustración 14).

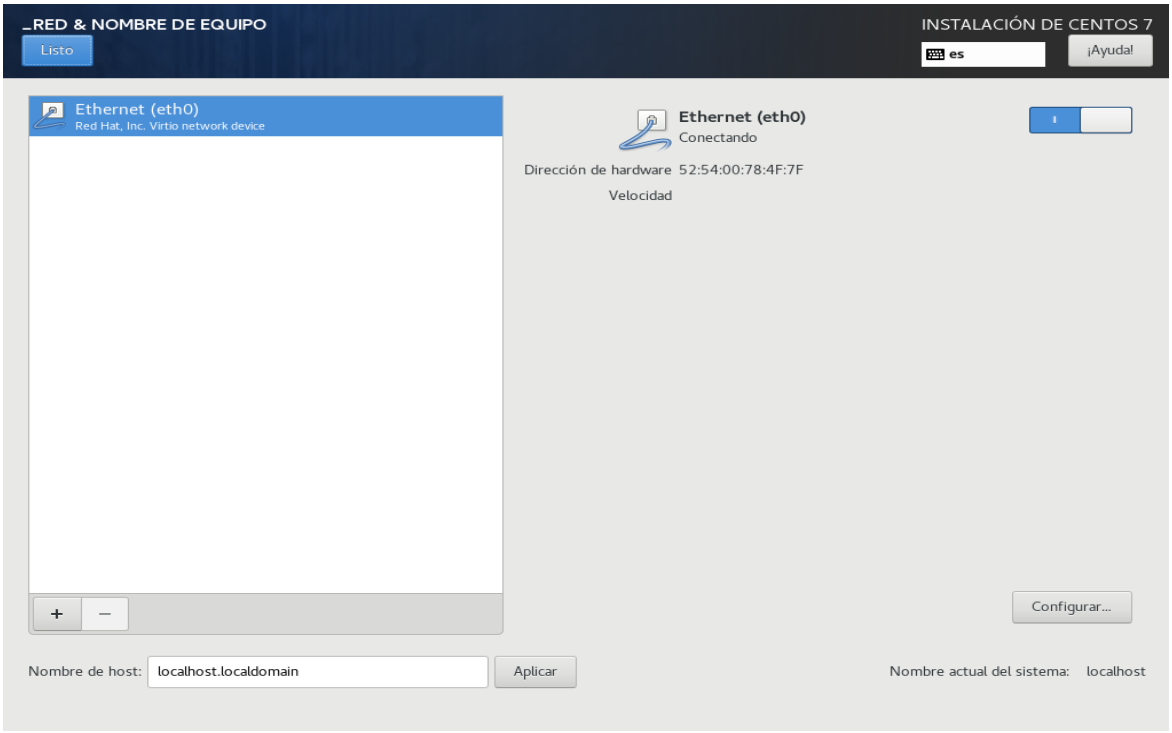


Ilustración 14: Proceso de instalación de CentOS 7 - Configuración de red

Seguidamente, se configuró el apartado “Fecha y Hora” para establecer la fecha y hora del sistema, en este caso, por red (Ilustración 15).

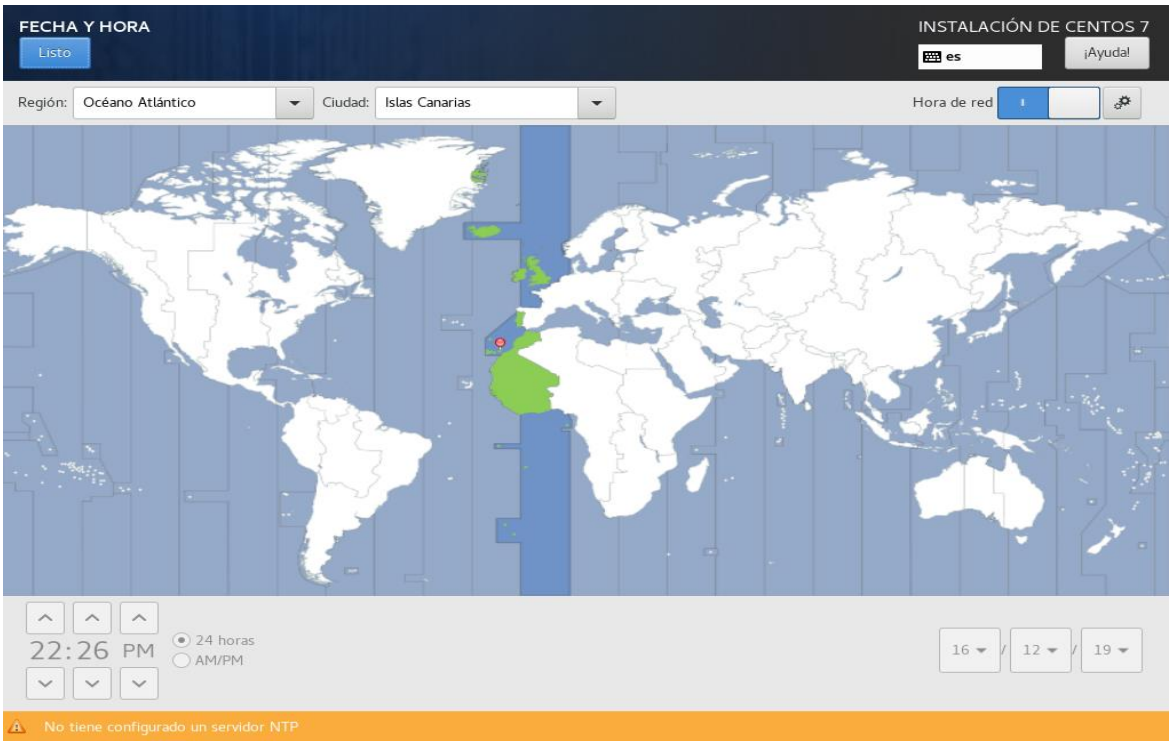


Ilustración 15: Proceso de instalación de CentOS 7 - Configuración de fecha y hora

Luego, se configuró el apartado "Selección de Software" para instalar con el sistema operativo el entorno gráfico GNOME (Ilustración 16).

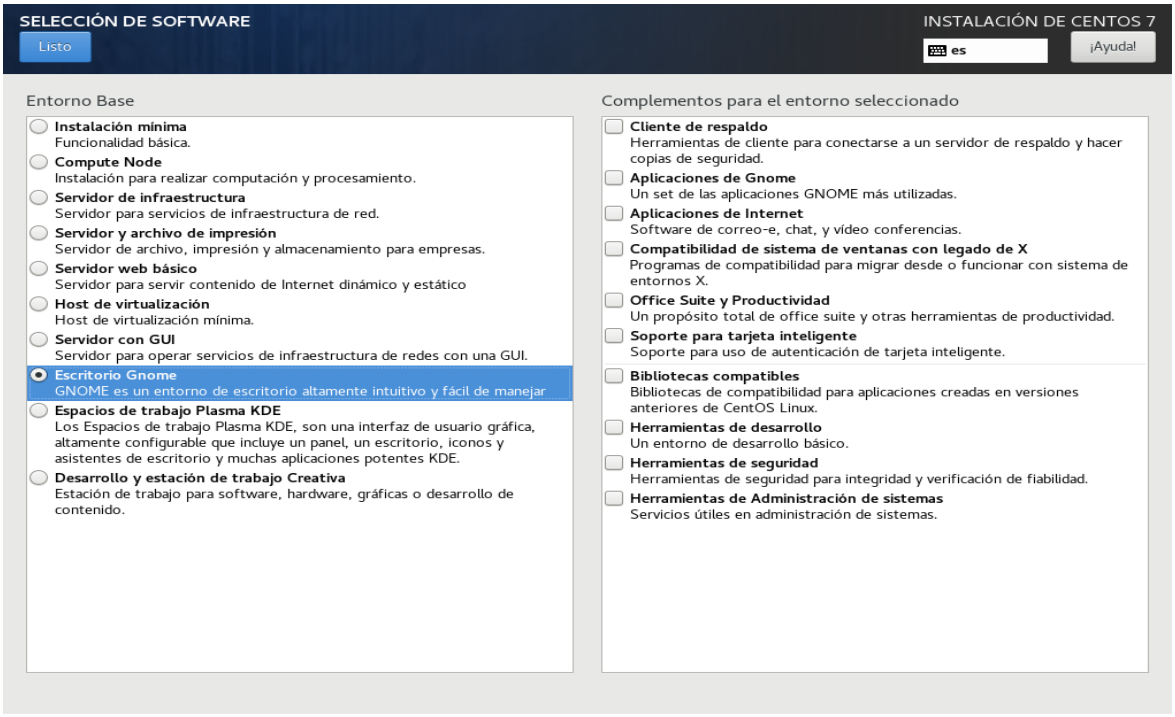


Ilustración 16: Proceso de instalación de CentOS 7 - Selección de software

Adicionalmente, se configuró el apartado "Destino de la Instalación" para seleccionar los dispositivos en los que se instalaría el sistema operativo y configurar las particiones de forma manual (Ilustración 17).

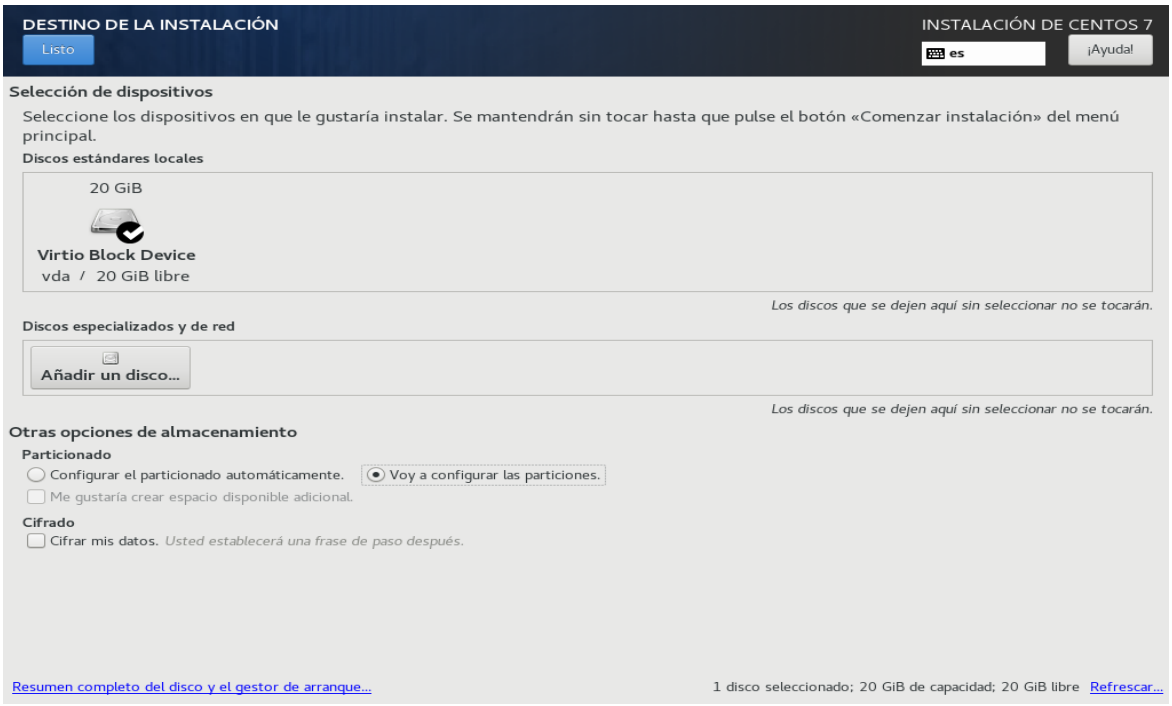


Ilustración 17: Proceso de instalación de CentOS 7 - Selección de destino de instalación

Al configurar las particiones manualmente se requirió crear, en este caso de forma automática, los puntos de montaje necesarios para la instalación (Ilustración 18).

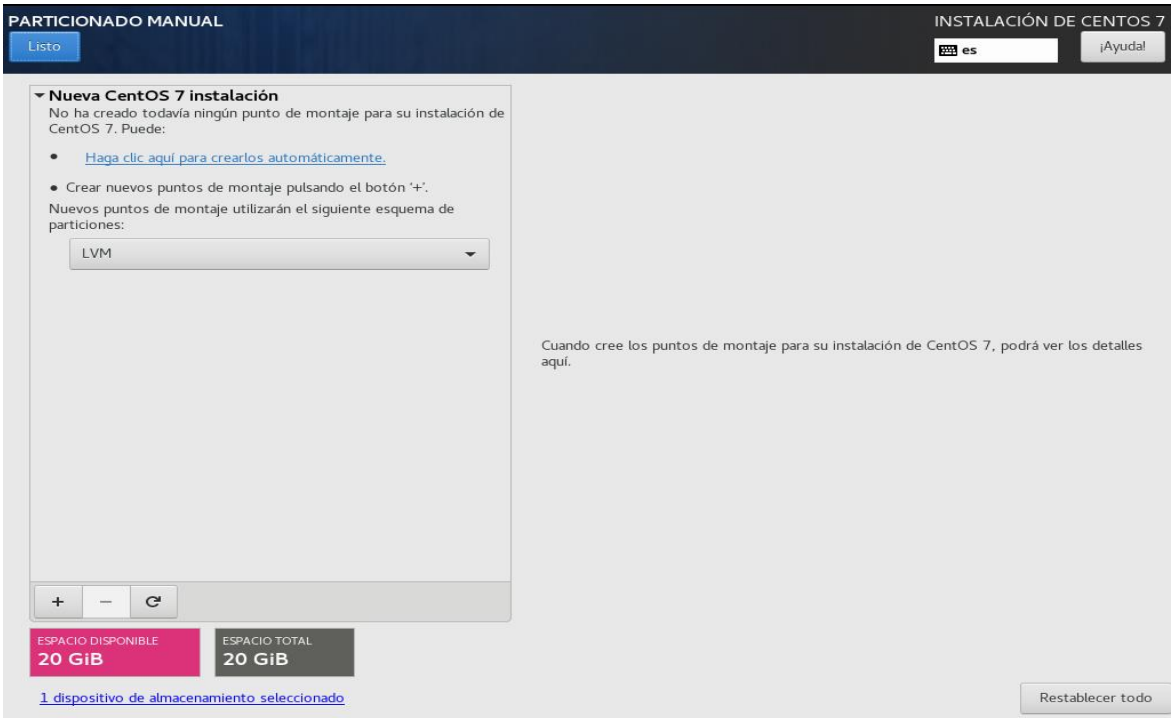


Ilustración 18: Proceso de instalación de CentOS 7 - Particionado automático I

Una vez creados los puntos de montaje, se le asignó a cada uno un espacio en disco adecuado, dependientemente del servidor físico en cuestión y la capacidad de almacenamiento de sus discos. Sin embargo, en todos los casos se le asignó un espacio de disco mayor al punto de montaje raíz del sistema (Ilustración 19).

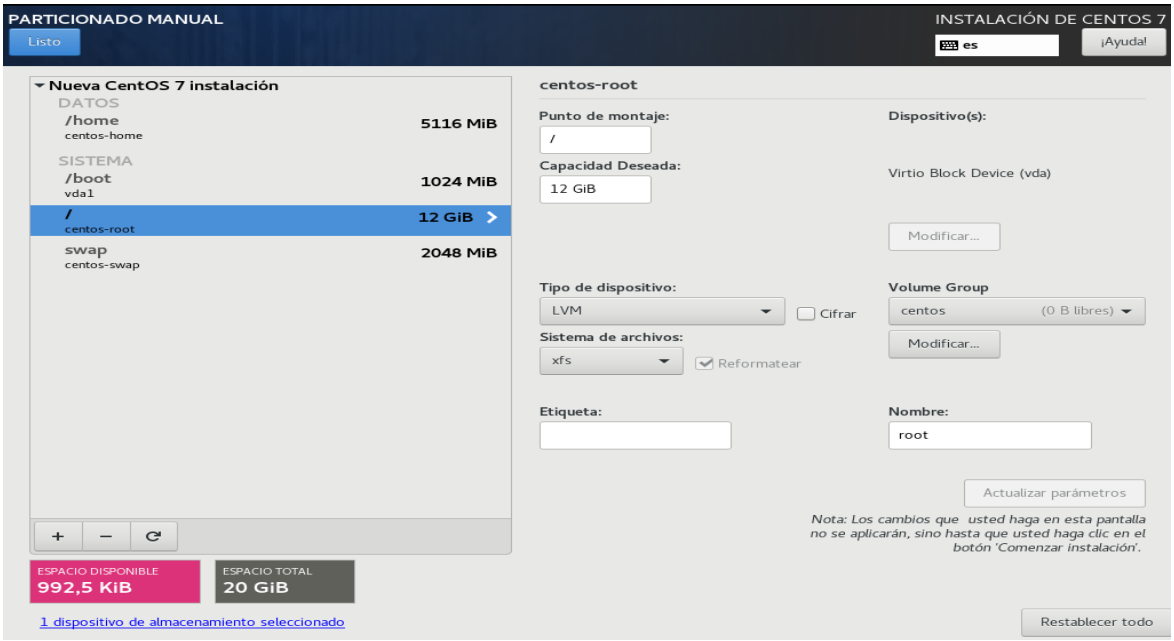


Ilustración 19: Proceso de instalación de CentOS 7 - Particionado automático II

Una vez se configuró el particionado, se aceptaron y guardaron los cambios, que se realizarían al comenzar la instalación (Ilustración 20).

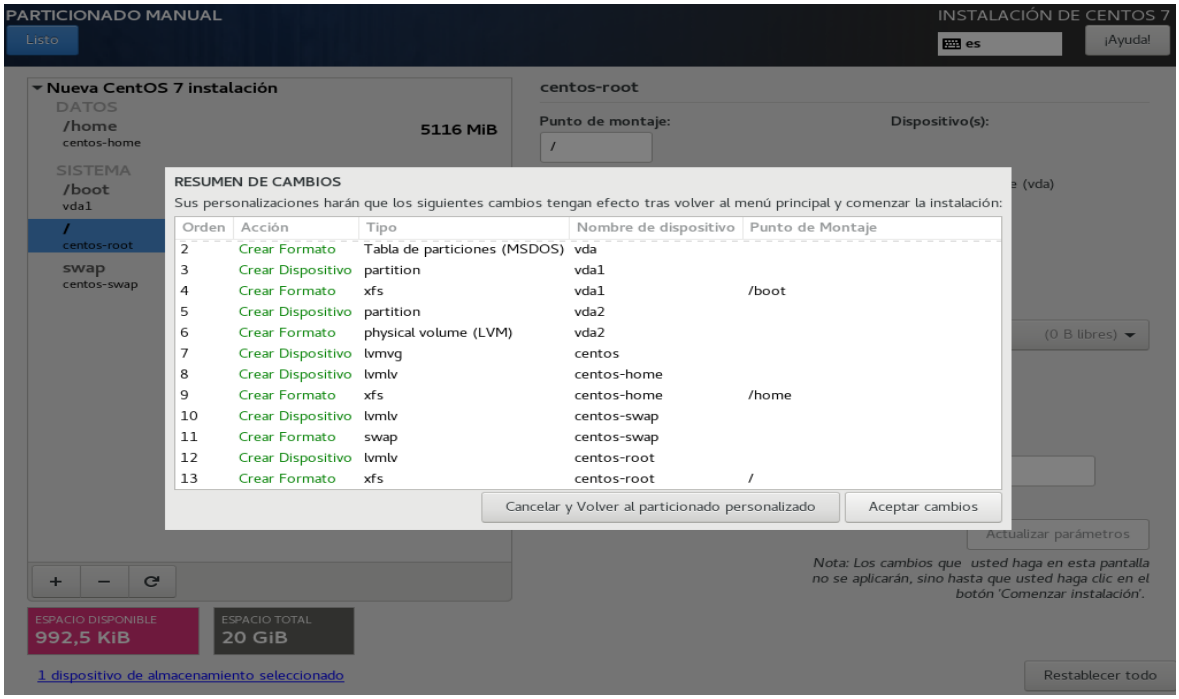


Ilustración 20: Proceso de instalación de CentOS 7 - Particionado automático III

Después de haber configurado todos los apartados de la instalación se procedió con el comienzo de esta (Ilustración 21).

Ilustración 21: Proceso de instalación de CentOS 7 - Resumen de configuración final

Por último, antes de finalizar la instalación, se estableció una contraseña para el usuario *root* y se prescindió de la creación de usuarios; después de la finalizar la instalación de reinició el sistema (Ilustración 22).

Ilustración 22: Proceso de instalación de CentOS 7 - Establecimiento de contraseña

ANEXO VI: CONFIGURACIÓN DEL SERVIDOR NG

Actualización del sistema

Debido a la reciente reinstalación del sistema operativo en el servidor NG fue necesario actualizar todos los paquetes del nuevo sistema:

```
# yum update -y
```

Instalación de wget

Para poder descargar la distribución de XAMPP desde la interfaz de línea de comandos primero fue necesario instalar la herramienta *wget*:

```
# yum install wget -y
```

Instalación de XAMPP

Por motivos de compatibilidad, primero se averiguó la versión de la distribución de XAMPP instalada en el servidor MR. Tras esto, se obtuvo de la página oficial de Apache Friends un instalador de la misma versión:

```
# wget RUTA_WEB_INSTALADOR_XAMPP
```

Una vez descargado el fichero, se le otorgó el permiso de ejecución:

```
# chmod +x FICHERO_INSTALADOR_XAMPP
```

Por último, se ejecutó el instalador:

```
# ./FICHERO_INSTALADOR_XAMPP
```

Una vez finalizada la instalación de la distribución se procedió a su configuración.

Modificación del fichero /opt/lampp/etc/extra/httpd-xampp.conf

Se modificó el fichero */opt/lampp/etc/extra/httpd-xampp.conf* para permitir a todos el acceso a la distribución de XAMPP.

Para ello, fue necesario comentar en dicho archivo la línea '*Require local'* y añadir la línea '*Require all granted'*:

```
…
# Require local
Require all granted
~
"/opt/lampp/etc/extra/httpd-xampp.conf"
```

Modificación del fichero /opt/lampp/etc/php.ini

Se modificó el fichero */opt/lampp/etc/php.ini* para cambiar el valor por defecto de algunos parámetros de PHP.

En este caso, se cambió el valor de los parámetros *post_max_size* y *upload_max_filesize* a 1024 MB para poder subir ficheros (generalmente de tipo SQL) con un tamaño máximo de 1GB al sistema de bases de datos del servidor NG:

```
…
post_max_size=1024M
…
upload_max_filesize=1024M
…
~
"/opt/lampp/etc/php.ini"
```

Modificación del fichero /opt/lampp/phpmyadmin/config.inc.php

Se modificó el fichero */opt/lampp/phpmyadmin/config.inc.php* para cambiar el valor por defecto de algunos parámetros de phpMyAdmin.

En primer lugar, y con el fin de obtener mayor seguridad en el acceso a phpMyAdmin se estableció el método de autenticación basado en *cookies*. Para ello, se cambió el valor del parámetro *$cfg['Servers'][$i]['auth_type']* (por defecto, '*config*') al valor '*cookie*':

```
…
$cfg['Servers'][$i]['auth_type'] = 'cookie';
…
~
"/opt/lampp/phpmyadmin/config.inc.php"
```

Con el fin de obtener una mayor seguridad en la autenticación mediante nombre de usuario y contraseña se cambió el valor del parámetro *$cfg['blowfish_secret']* (por defecto, '*xampp*') sustituyéndolo por una clave simétrica generada aleatoriamente mediante el método de cifrado *Blowfish*:

```
…
$cfg['blowfish_secret'] = 'CLAVE_BLOWFISH';
…
~
"/opt/lampp/phpmyadmin/config.inc.php"
```

Con el fin de obtener una mayor seguridad en la autenticación del usuario *root* se cambió el valor del parámetro *$cfg['Servers'][$i]['password']* (por defecto con un valor vacío) añadiendo la contraseña deseada para dicho usuario:

```
…
$cfg['Servers'][$i]['password'] = 'CONTRASEÑA_ROOT';
…
~
"/opt/lampp/phpmyadmin/config.inc.php"
```

Con el fin de obtener una mayor seguridad en la autenticación del usuario *pma* se cambió el valor del parámetro *$cfg['Servers'][$i]['controlpass']* (por defecto con un valor vacío) añadiendo la contraseña deseada para dicho usuario:

```
…
$cfg['Servers'][$i]['controlpass'] = '***CONTRASEÑA_PMA***';
…
~
"/opt/lampp/phpmyadmin/config.inc.php"
```

Por último y con el fin de habilitar el almacenamiento de archivos temporales del servidor web se añadió el parámetro *$cfg['TempDir']* con el valor *'/tmp*':

```
…
$cfg['TempDir'] = '/tmp';
…
~
"/opt/lampp/phpmyadmin/config.inc.php"
```

Automatización del inicio de XAMPP

Para facilitar el inicio de todos los componentes de la distribución de XAMPP se decidió automatizarlo de forma que este se ejecutara justo después del arranque del servidor.

Para ello, primero se creó un enlace simbólico del *script* de ejecución de XAMPP en el directorio */etc/init.d/* :

```
# ln -s /opt/lampp/lampp /etc/init.d/lampp
```

Y a continuación, se añadió el *script* a la gestión de la herramienta *chkconfig*, que se encargaría de iniciar el *script* con cada arranque del servidor NG:

```
# chkconfig --add lampp
```

Habilitación de servicio mediante FirewallD: Servicio HTTP

Para permitir las conexiones entrantes al servidor web primero fue necesario habilitar, mediante una regla, el protocolo HTTP (puerto 80) en el cortafuegos del sistema a través de *FirewallD*:

```
# firewall-cmd --zone=public --add-service=http --permanent
```

Además, y debido a que la regla se añadió de forma permanente, fue necesario recargar el sistema FirewallD para hacer efectivos los cambios:

```
# firewall-cmd --reload
```

Inicio de XAMPP

A continuación, se inició la distribución de XAMPP en el servidor NG:

```
# /opt/lampp/lampp start
```

Modificación de las tablas de privilegios MySQL mediante phpMyAdmin

Una vez iniciado el servidor web, fue necesario establecer las mismas contraseñas especificadas en el fichero */opt/lampp/phpmyadmin/config.inc.php* en las tablas de privilegios MySQL de los usuarios *root* y *pma*.

En primer lugar, se accedió a la página de autenticación de la interfaz web phpMyAdmin del servidor (Ilustración 23).

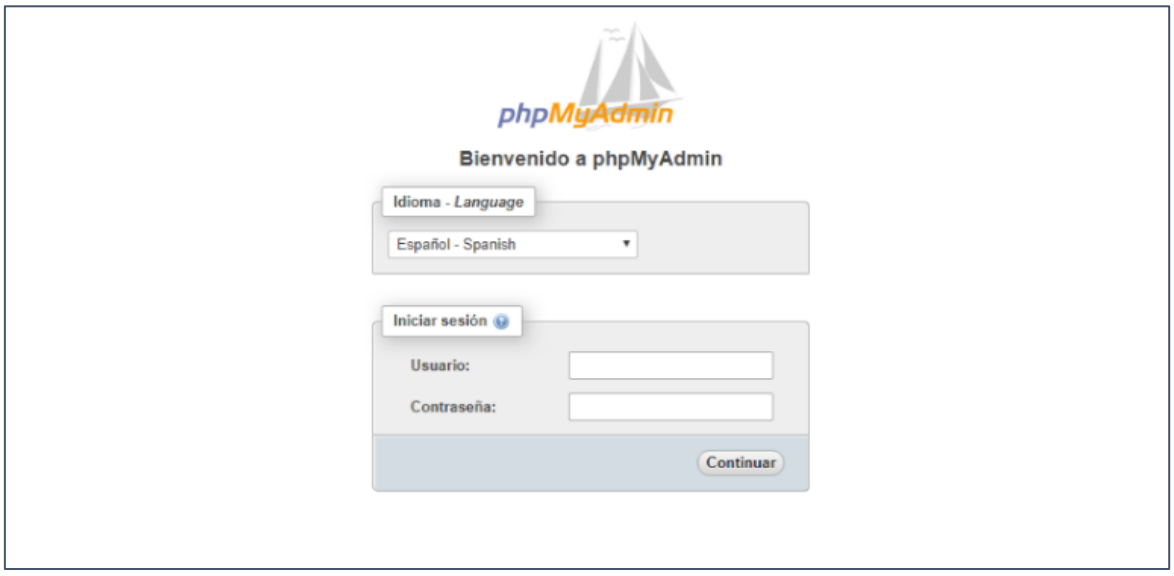


Ilustración 23: Página de autenticación de phpMyAdmin

En la primera autenticación fue necesario introducir el nombre de usuario administrador *root* con una contraseña vacía, ya que los usuarios existentes no tenían establecida ninguna contraseña en sus respectivas tablas de privilegios MySQL.

Una vez autenticado el usuario se mostró la página de inicio de phpMyAdmin (Ilustración 24).

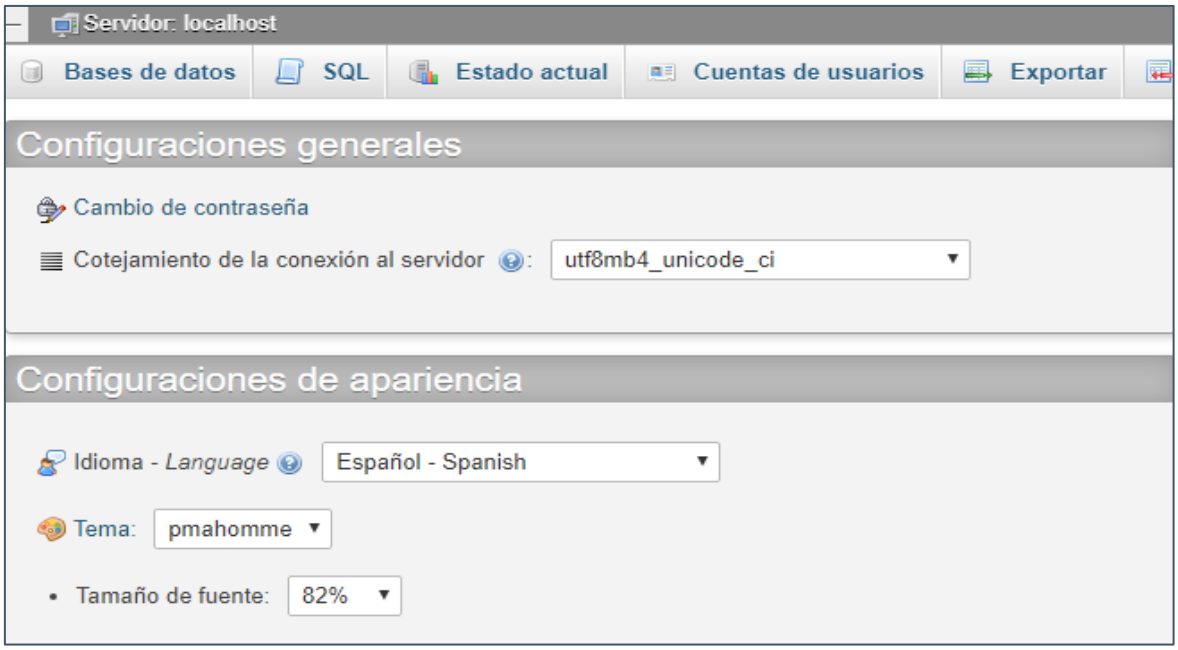


Ilustración 24: Página de inicio de phpMyAdmin

Seguidamente, se accedió a la vista global de las cuentas de los usuarios existentes mediante la pestaña "Cuentas de usuarios" (Ilustración 25).

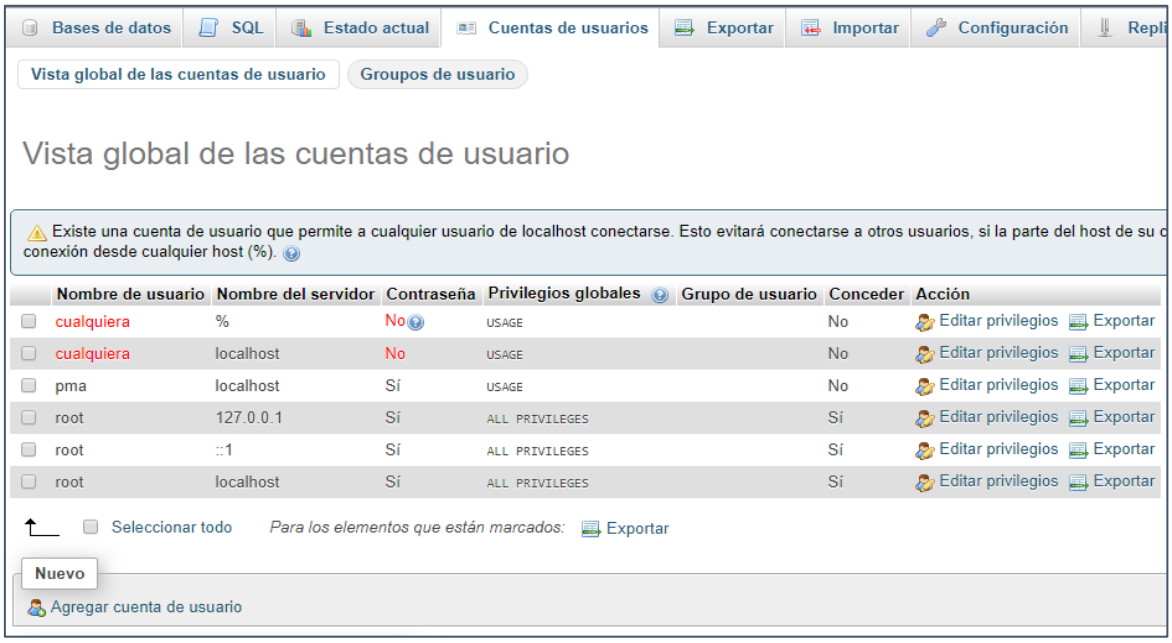


Ilustración 25: Vista global de las cuentas de usuarios de phpMyAdmin

Para establecer las contraseñas de los usuarios *root* y *pma* se accedió a la pantalla de privilegios de cada usuario mediante el enlace "Editar privilegios" (Ilustración 26).

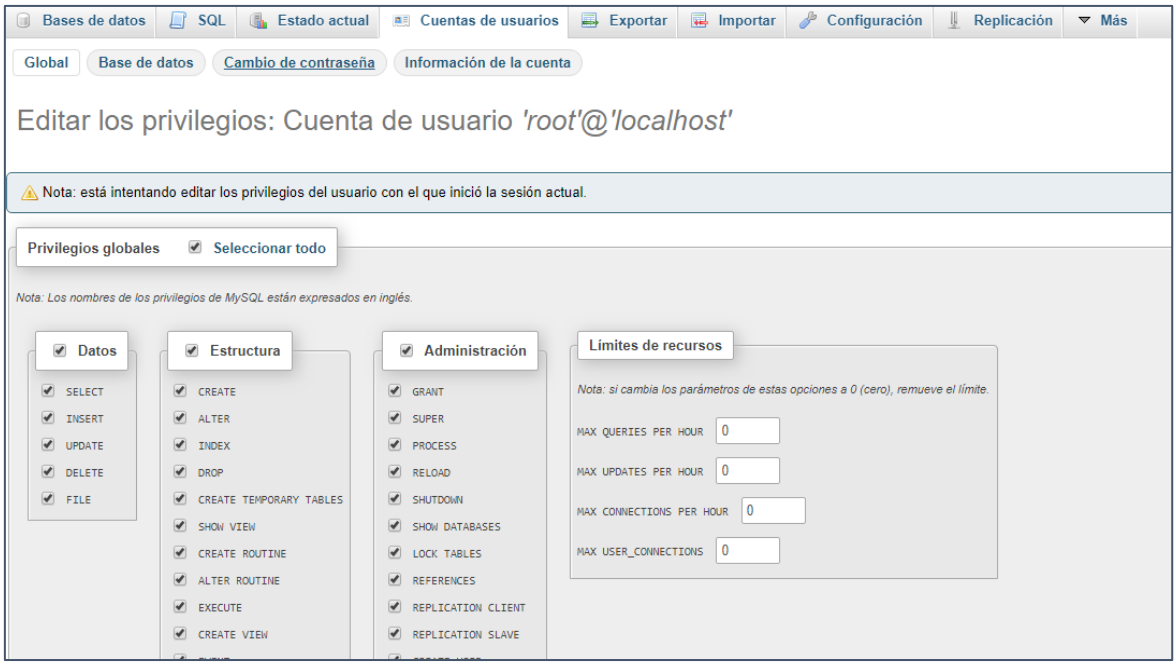


Ilustración 26: Página de modificación de privilegios de usuario de phpMyAdmin

Finalmente, se accedió a la pantalla de cambio de contraseña mediante la pestaña "Cambio de contraseña" del submenú desplegado, en la cual se estableció y guardó la nueva contraseña (Ilustración 27).

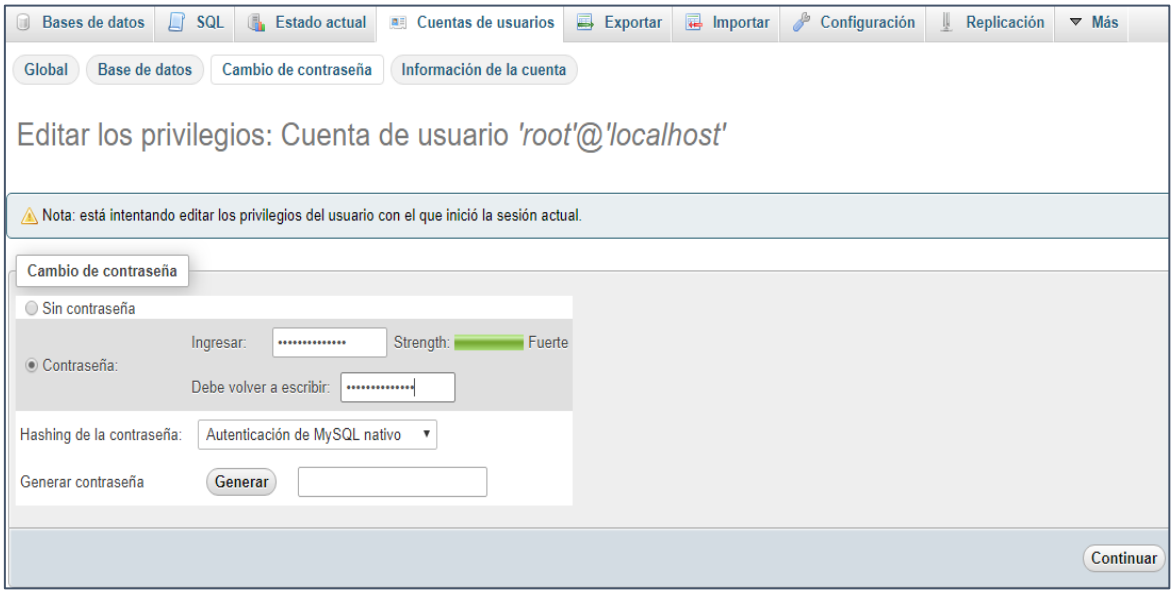


Ilustración 27: Página de modificación de privilegios de usuario de phpMyAdmin - Cambio de contraseña

Comprobación de seguridad de XAMPP

Con el objetivo de mejorar la seguridad del servidor web se ejecutó el siguiente comando:

```
# /opt/lampp/lampp security
```

Mediante este comando se pudieron realizar varias comprobaciones de seguridad con las que se pudo asegurar que:

- El sistema de bases de datos MySQL/MariaDB no fuera accesible mediante la red.
- El usuario de MySQL/MariaDB *root* tuviera especificada una contraseña.
- El usuario de MySQL/MariaDB *pma* tuviera especificada una contraseña.
- El servidor FTP, denominado ProFTPD, tuviera una contraseña especificada distinta a la establecida por defecto.

Reinicio de XAMPP

Por último, se reinició la distribución de XAMPP:

```
# /opt/lampp/lampp restart
```

ANEXO VII: MIGRACIÓN TEMPORAL DE SERVICIOS WEB DEL SERVIDOR MR AL SERVIDOR NG

Migración del fichero /opt/lampp/etc/extra/httpd-vhosts.conf

Como paso previo y con el fin de reproducir el mismo espacio de alojamiento compartido entre servicios, se copió el fichero de alojamiento compartido */opt/lampp/etc/extra/httpd-vhosts.conf* del servidor MR al servidor NG utilizando la herramienta *scp* de transferencia remota de ficheros:

```
# scp /opt/lampp/etc/extra/httpd-vhosts.conf
root@DIRECCION_IP_SERVIDOR_NG:/opt/lampp/etc/extra/
```

Deshabilitación del servicio web en el servidor original

Para migrar un servicio web antes fue necesario asegurarse de que este fuera inaccesible mientras el proceso era llevado a cabo para evitar, sobre todo, errores en las posibles bases de datos asociadas.

Para ello y en primer lugar, se detuvo la distribución de XAMPP del servidor MR:

```
# /opt/lampp/lampp stop
```

Una vez detenida la distribución, se movió el directorio contenedor de los documentos HTML del servicio que se iba a migrar a un directorio superior:

```
# mv /opt/lampp/htdocs/DIRECTORIO_PROYECTO_WEB /opt/lampp/
```

Teniendo en cuenta que todos y cada uno de los servicios web del servidor MR tenía sus documentos HTML localizados en el directorio raíz de alojamiento */opt/lampp/htdocs/*, el traslado de los documentos HTML del servicio a un directorio superior, fuera del directorio de alojamiento, bastó para que este dejara de estar disponible.

Seguidamente, se volvió a iniciar la distribución de XAMPP en el servidor MR:

```
# /opt/lampp/lampp start
```

Esta forma de deshabilitar cada servicio fue posible debido a que los servicios que se iban a migrar no tenían dependencias con otros externos.

Migración de los documentos HTML del servicio web

Para transferir de forma remota el directorio contenedor de los documentos HTML de cada servicio web del servidor MR al servidor NG se ejecutó, en este último, el siguiente comando:

```
# scp -r root@DIRECCION_IP_SERVIDOR_MR:/opt/lampp/DIRECTORIO_PROYECTO_WEB/ /opt/lampp/htdocs/
```

Migración de los datos de las bases de datos del servicio web

También fue necesario exportar las bases de datos de aquellos servicios que lo requerían. Para exportar las bases de datos asociadas a cada servicio web se ejecutó, por cada base de datos asociada, el siguiente comando en el servidor MR:

```
# /opt/lampp/bin/mysqldump -u NOMBRE_USUARIO_BD -p NOMBRE_BD > NOMBRE_FICHERO_BD.sql
```

De esta forma, se guardaron los datos de cada base de datos del servicio que se iba a migrar en un fichero SQL independiente, en este caso, creado en el directorio */root* del servidor MR.

Tras esto, se transfirieron de forma remota todos los ficheros SQL generados al servidor NG. Para ello se ejecutó, por cada fichero generado, el siguiente comando en dicho servidor:

```
# scp root@DIRECCION_IP_SERVIDOR_MR:/root/NOMBRE_FICHERO_BD.sql /root/
```

Luego, fue necesario crear en el servidor NG una base de datos por cada fichero SQL transferido del servidor MR. Para ello, por cada fichero, se siguió el siguiente proceso:

1) Se accedió a la interfaz del sistema de bases de datos, en este caso MariaDB, con el siguiente comando:

   ```
   # /opt/lampp/bin/mysql -u NOMBRE_USUARIO_BD -p
   ```

2) Dentro de la interfaz, se creó la base de datos correspondiente, con el mismo nombre que su base de datos original:

   ```
   MariaDB [(none)]> CREATE DATABASE NOMBRE_BD;
   ```

3) E, inmediatamente, se finalizó la conexión con la interfaz:

```
MariaDB [(none)]> exit;
```

Finalmente, se importaron en el servidor NG los datos de cada fichero SQL en su base de datos correspondiente ejecutando, por cada fichero SQL, el siguiente comando:

```
# /opt/lampp/bin/mysql -u NOMBRE_USUARIO_BD -p -h localhost NOMBRE_BD < NOMBRE_BD.sql
```

Reinicio de XAMPP

Tras migrar los documentos HTML y las bases de datos asociados del servicio web al servidor NG se reinició la distribución de XAMPP en este último:

```
# /opt/lampp/lampp restart
```

Migración de los nombres de dominio de un servicio web mediante ULPnet

Por último, para permitir el fácil acceso a cada servicio web alojado en el servidor NG fue necesario desasociar sus nombres de dominio del servidor MR y asociarlos al servidor NG a través de la herramienta de gestión de redes de ULPnet.

ANEXO VIII: CONFIGURACIÓN DEL SERVIDOR MR

Actualización del sistema

En primer lugar, se actualizaron todos los paquetes del sistema recién instalado:

```
# yum update -y
```

Establecimiento de un enrutamiento simétrico para las interfaces de red

Con el fin de permitir a los equipos de la red interna acceder a los servicios del servidor MR a través de su dirección IP pública fue necesario configurar las interfaces de red de forma que el enrutamiento de paquetes se hiciera de forma simétrica. Para esto se consultó un artículo de Jens Depuydt [41].

En primer lugar, fue necesario establecer una dirección IP estática a la interfaz de red conectada a la red interna (*em2*). Para ello se especificaron en el fichero */etc/sysconfig/network-scripts/ifcfg-em2* del servidor los siguientes parámetros:

```
…
BOOTPROTO=none
…
ONBOOT=yes
IPADDR=DIRECCIÓN_IP_PRIVADA_SERVIDOR_MR
PREFIX=PREFIJO_MÁSCARA_SUBRED
GATEWAY=DIRECCIÓN_IP_PUERTA_DE_ENLACE_RED_PRIVADA
DNS1=193.145.138.100
DNS2=193.145.138.200
~
“/etc/sysconfig/network-scripts/ifcfg-em2”
```

Debido a que el servidor DHCP de la red pública del centro se encargaba siempre de asignar de forma estática una dirección IP pública al servidor MR no fue necesario configurar la interfaz de red conectada a dicha red (*em1*).

A continuación, se creó un fichero de configuración de enrutamiento */etc/sysconfig/network-scripts/route-em1* de la interfaz de la red pública (*em1*) con el siguiente contenido:

```
DIRECCIÓN_IP_RED_PÚBLICA/MÁSCARA_SUBRED dev em1 tab 1
default via DIRECCIÓN_IP_PUERTA_ENLACE_RED_PÚBLICA dev em1 tab 1
~
"/etc/sysconfig/network-scripts/route-em1"
```

Asimismo, se creó el fichero de configuración de enrutamiento */etc/sysconfig/network-scripts/route-em2* de la interfaz de la red privada (*em2*) con el siguiente contenido:

```
DIRECCIÓN_IP_RED_PRIVADA/MÁSCARA_SUBRED dev em2 tab 2
default via DIRECCIÓN_IP_PUERTA_ENLACE_RED_PRIVADA dev em2 tab 2
~
"/etc/sysconfig/network-scripts/route-em2"
```

Además, se creó un fichero de configuración de las reglas de enrutamiento */etc/sysconfig/network-scripts/rule-em1* de la interfaz de la red pública (*em1*) con el siguiente contenido:

```
from DIRECCIÓN_IP_PÚBLICA_SERVIDOR/MASC_SUBRED tab 1 priority 100
~
"/etc/sysconfig/network-scripts/rule-em1"
```

Seguidamente, se creó el fichero de configuración de las reglas de enrutamiento */etc/sysconfig/network-scripts/rule-em2* de la interfaz de la red privada (*em2*) con el siguiente contenido:

```
from DIRECCIÓN_IP_PRIVADA_SERVIDOR/MASC_SUBRED tab 2 priority 200
~
"/etc/sysconfig/network-scripts/rule-em2"
```

Por último, se reinició el servicio de red:

```
# systemctl restart network
```

Deshabilitación de FirewallD

Debido a que en el servidor se gestionarían conexiones con varias redes (red pública, red privada y red virtual) se necesitaría un control más exhaustivo en las reglas del cortafuegos por lo que se decidió emplear el sistema de gestión de cortafuegos clásico *iptables* en vez del instalado por defecto *FirewallD*.

De esta forma, se detuvo el servicio de *Firewalld*:

```
# systemctl stop firewalld
```

Se deshabilitó su inicio automático:

```
# systemctl disable firewalld
```

Y se enmascaró para prevenir que fuera iniciado por otros servicios:

```
# systemctl mask --now firewalld
```

Habilitación de Iptables

A continuación, se instaló *iptables*:

```
# yum install iptables-services -y
```

Se habilitó su inicio automático:

```
# systemctl enable iptables
```

Y se inició el servicio de *iptables*:

```
# systemctl start iptables
```

Preconfiguración de Iptables

En primer lugar se cambiaron las políticas de aceptación de paquetes tanto de las conexiones entrantes (*input*) como de las conexiones reenviadas (*forward*) al modo más restrictivo (*drop*):

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
```

Tras esto, se hicieron efectivos los cambios en el sistema con el siguiente comando:

```
# /sbin/service iptables save
```

Seguidamente se añadieron varias reglas al sistema de cortafuegos que permitieran las conexiones más básicas al servidor como SSH, HTTP y HTTPS, de forma que el fichero de configuración */etc/sysconfig/iptables* tuviera el siguiente contenido:

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [30:3059]
-A INPUT -i lo -m state --state NEW -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -m state --state NEW -j ACCEPT
-A INPUT -p udp -m udp -m multiport --dports 80,443 -m state --state NEW -j ACCEPT
-A INPUT -p tcp -m tcp -m multiport --dports 22,80,443 -m state --state NEW -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
…
~
"/etc/sysconfig/iptables"
```

Por último, se reinició el servicio de *iptables*:

```
# systemctl restart iptables
```

Instalación del repositorio EPEL

Se requirió instalar el repositorio EPEL como prerrequisito para las subsiguientes instalaciones en el servidor:

```
# yum install epel-release -y
```

Habilitación de puerto mediante Iptables: Puerto 3389/tcp

Se configuró el sistema cortafuegos para permitir las conexiones RDP al servidor desde los equipos de la red interna del centro. Para ello, fue necesario habilitar el puerto 3389 en el cortafuegos de forma que el fichero de configuración */etc/sysconfig/iptables* tuviera el siguiente contenido:

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [30:3059]
-A INPUT -i lo -m state --state NEW -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -m state --state NEW -j ACCEPT
-A INPUT -p udp -m udp -m multiport --dports 80,443 -m state --state NEW -j ACCEPT
-A INPUT -p tcp -m tcp -m multiport --dports 22,80,443 -m state --state NEW -j ACCEPT
-A INPUT -i em2 -p tcp -m tcp --dport 3389 -m state --state NEW -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
…
~
“/etc/sysconfig/iptables”
```

Una vez guardados los cambios se reinició el servicio de *iptables*:

```
# systemctl restart iptables
```

Instalación de XRDP

En primer lugar, se decidió instalar el servidor XRDP con el fin de facilitar y agilizar tanto la configuración como la gestión del servidor.

Esta instalación se realizó de acuerdo con la guía de instalación proporcionada por el portal web IT’zGeek [42].

En primer lugar, se instalaron los paquetes de XRDP:

```
# yum install xrdp tigervnc-server -y
```

Se inició el servicio XRDP:

```
# systemctl start xrdp
```

Se automatizó el inicio de XRDP:

```
# systemctl enable xrdp
```

Se ajustaron los contextos de seguridad de los ficheros de XRDP mediante SELinux:

```
# chcon --type=bin_t /usr/sbin/xrdp
# chcon --type=bin_t /usr/sbin/xrdp-sesman
```

Luego, se comprobó la conexión remota del servidor XRDP. Para ello desde un equipo de la red interna (con Windows) se abrió el programa "Conexión a Escritorio Remoto" y se conectó al servidor XRDP introduciendo su dirección IP (Ilustración 28).

Ilustración 28: Ventana de Conexión a Escritorio Remoto de Windows

Tras esto, se ignoraron las advertencias de error de certificado (Ilustración 29).

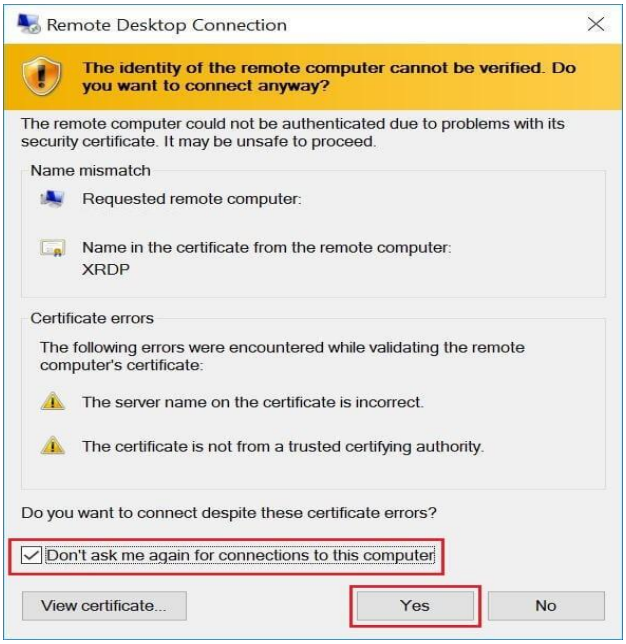


Ilustración 29: Ventana de advertencia de error de certificado del servidor XRDP

Por último, utilizando el módulo "Xvnc" y unas credenciales válidas, se accedió al escritorio del servidor XRDP (Ilustración 30).

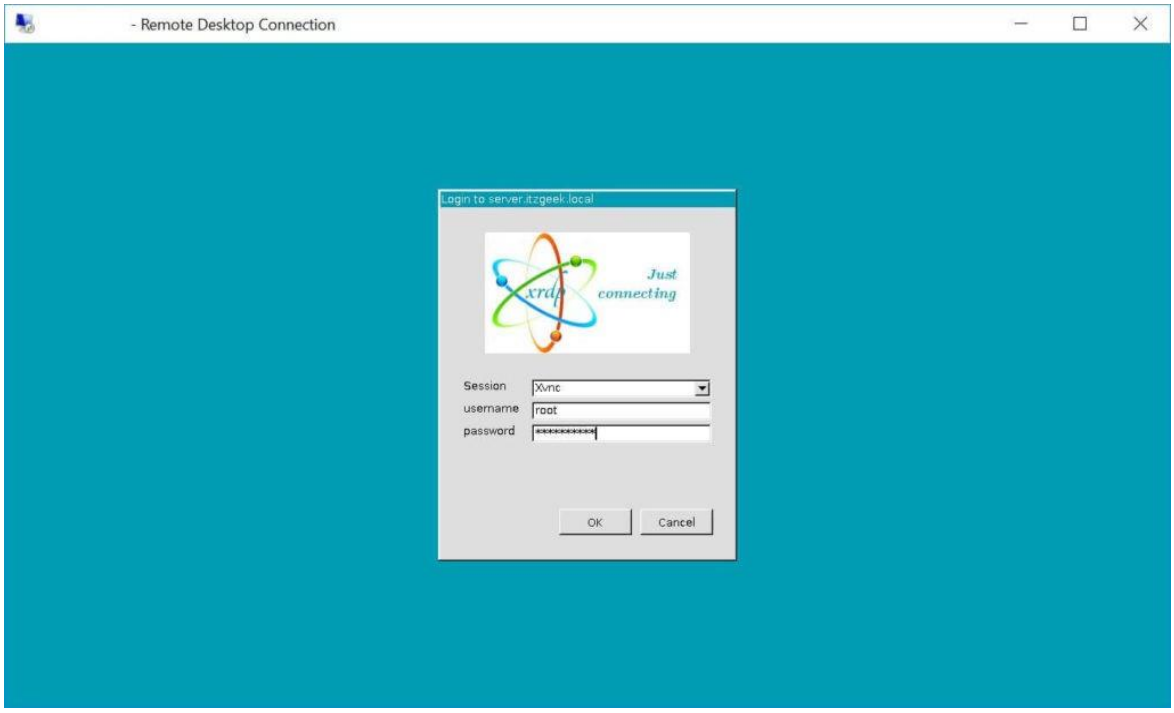


Ilustración 30: Página de acceso a escritorio del servidor XRDP

Instalación de KVM

Tras la instalación de XRDP se procedió a instalar el hipervisor de KVM para poder crear y gestionar los servidores virtuales que compondrán la nueva infraestructura.

En primer lugar, se comprobó que el hardware de la máquina en la que se iba a instalar la herramienta ofrecía soporte para virtualización:

```
# egrep -c '(svm|vmx)' /proc/cpuinfo
```

Con este comando se pudo acceder al fichero en el que se encontraba toda la información relativa a la CPU para contar el número de coincidencias encontrados de las banderas *svm* (AMD-V, de AMD) y *vmx* (VT-X, de Intel), indicando si existía (por cada procesador) soporte para la virtualización o no. Como el número de coincidencias fue igual al número de procesadores se pudo concluir que efectivamente había soporte para la virtualización.

Seguidamente, se comprobó que la virtualización estuviera activada desde la BIOS.

Se instalaron los paquetes de KVM:

```
# yum install qemu-kvm qemu-img virt-manager libvirt libvirt-python libvirt-client virt-install virt-viewer bridge-utils -y
```

Se inició el servicio del hipervisor, *libvirtd*:

```
# systemctl start libvirtd
```

Se automatizó el inicio del hipervisor*:*

```
# systemctl enable libvirtd
```

Se comprobó que el módulo de KVM estuviera cargado en el sistema:

```
# lsmod | grep kvm
```

Por último, se reinició el servidor MR para que se hicieran efectivos los cambios realizados durante la instalación de KVM:

```
# reboot
```

Preconfiguración de KVM

Con el fin de facilitar y ahorrar tiempo en la creación de futuras máquinas virtuales mediante KVM se decidió crear, bajo el directorio */var/lib/libvirt/*, dos subdirectorios.

Se creó el subdirectorio */var/lib/libvirt/iso/*, para el almacenamiento de imágenes ISO correspondientes a sistemas operativos:

```
# mkdir /var/lib/libvirt/iso/
```

Se creó el subdirectorio */var/lib/libvirt/templates/*, para el almacenamiento de imágenes de discos virtuales ya configurados como plantillas para la creación de nuevas máquinas virtuales:

```
# mkdir /var/lib/libvirt/templates/
```

Habilitación de servicio mediante FirewallD: Servicio HTTP

Además, se configuró el sistema de cortafuegos para permitir las conexiones HTTP desde los equipos de la red externa del centro. Para ello, fue necesario habilitar el protocolo HTTP en la zona externa del cortafuegos de la siguiente manera:

```
# firewall-cmd --zone=external --add-service=http --permanent
# firewall-cmd --reload
```

Instalación de NGINX

A continuación, se instaló el proxy inverso de NGINX, el cual permitiría a los clientes acceder a los servidores virtuales.

En primer lugar, se instaló el paquete de NGINX:

```
# yum install nginx -y
```

Seguidamente, se inició el servicio de NGINX:

```
# systemctl start nginx
```

Por último, se automatizó el inicio de NGINX:

```
# systemctl enable nginx
```

Preconfiguración de NGINX como proxy inverso

Tras haber instalado el proxy inverso se decidió configurarlo mediante el fichero */etc/nginx/nginx.conf* de la siguiente manera:

```
…
http {
        …
        proxy_buffers 32 4m;
        proxy_busy_buffers_size 25m;
        proxy_buffer_size 512k;
        proxy_ignore_headers "Cache-Control" "Expires";
        proxy_max_temp_file_size 0;
        proxy_set_header Host $host;
        proxy_set_header X-Real_IP $remote_addr;
        proxy_set_header X-Forwarded_For $proxy_add_x_forwarded_for;
        client_max_body_size 1024M;
        client_body_buffer_size 4m;
        proxy_connect_timeout 300;
        proxy_send_timeout 300;
        proxy_read_timeout 300;
        send_timeout 300;
        proxy_intercept_errors off;
        server_tokens off;

        include /etc/nginx/conf.d/*.conf;
}
…
~
“/etc/nginx/nginx.conf”
```

Una vez se realizaron y guardaron los cambios, se recargó el sistema de NGINX:

```
# nginx -s reload
```

Instalación de PIGZ

Finalmente, se requirió la instalación de la herramienta de compresión de ficheros PIGZ:

```
# yum install pigz -y
```

ANEXO IX: CREACIÓN DE SERVIDORES VIRTUALES EN EL SERVIDOR MR

Descarga de la imagen ISO del sistema operativo CentOS 7

Se decidió que el sistema operativo instalado en cada servidor virtual sería el sistema operativo CentOS 7 en su versión mínima.

Por ello, en primer lugar, se descargó de la página oficial de CentOS la imagen ISO de la versión mínima de dicho sistema operativo y fue situada en el directorio */var/lib/libvirt/iso/* del servidor MR.

Inicio del proceso de creación de una máquina virtual

Seguidamente, se ejecutó el Gestor de Máquinas Virtuales, el cual mostraría una lista (en ese momento vacía) de todas las máquinas virtuales existentes declaradas, tanto encendidas como apagadas.

Para comenzar la creación de un nuevo servidor virtual se pulsó el icono  mostrado en la esquina superior izquierda del gestor. Con esto se mostró un *wizard* que, dividido en una serie de pasos, sirvió de guía para la creación de una nueva máquina virtual.

Primer paso: Especificación del tipo de instalación del sistema operativo de la máquina virtual

El primer paso consistió en especificar el tipo de instalación del sistema operativo de la máquina virtual (Ilustración 31).

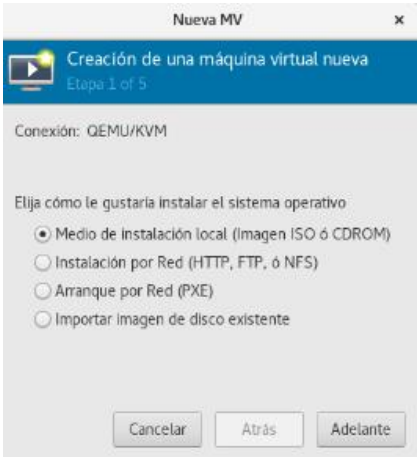


Ilustración 31: Creación de una máquina virtual de KVM - Etapa 1

De las múltiples opciones mostradas se escogió la denominada “Medio de instalación local”, ya que se utilizaría una imagen ISO para la instalación del sistema operativo.

Segundo paso: Especificación del volumen de almacenamiento como medio de instalación de la máquina virtual

El segundo paso consistió en especificar el volúmen de almacenamiento (una imagen ISO en este caso) como medio de instalación del sistema operativo de la nueva máquina virtual (Ilustración 32).

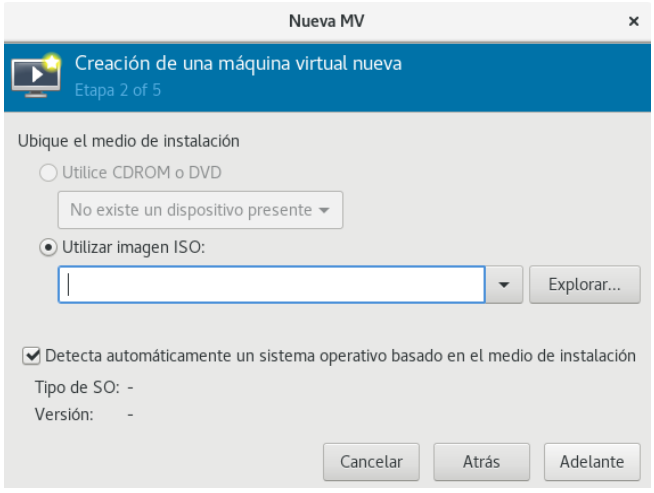


Ilustración 32: Creación de una máquina virtual de KVM - Etapa 2-1

Sin embargo, antes fue conveniente crear un nuevo *pool* de almacenamiento de tipo *directorio* dedicado a las imágenes ISO con destino en el anteriormente creado directorio */var/lib/libvirt/iso/*.

Una vez hecho esto, se seleccionó del *pool* creado el volumen correspondiente a la imagen ISO del sistema operativo CentOS 7 en su versión mínima (Ilustración 33).

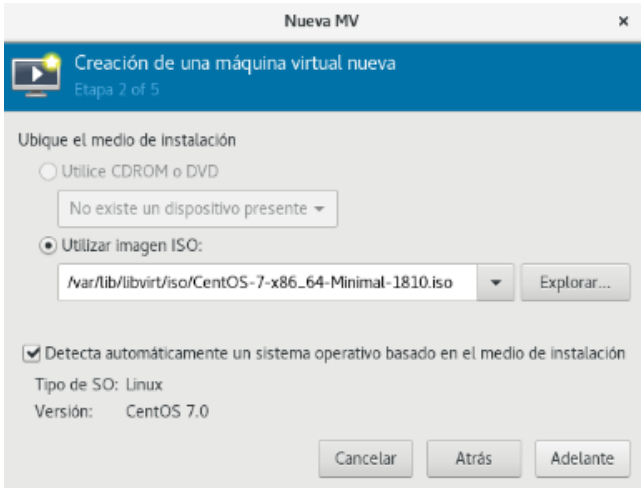


Ilustración 33: Creación de una máquina virtual de KVM - Etapa 2-2

Tercer paso: Especificación de la cantidad de memoria y el número de procesadores de la máquina virtual

El tercer paso consistió en especificar la cantidad de memoria (RAM) y el número de procesadores de los que dispondría la nueva máquina virtual, como se muestra en la Ilustración 34, en la cual se puede observar como ejemplo la asignación de 1 GB de memoria y 1 procesador. La especificación de estos parámetros fue adaptado a cada servidor virtual según la cantidad de recursos que necesitaría para ejecutar el servicio web que alojaría.

Ilustración 34: Creación de una máquina virtual de KVM - Etapa 3

Cuarto paso: Especificación del medio de almacenamiento de la máquina virtual

El cuarto paso consistió en la creación o asignación de un disco de almacenamiento para la nueva máquina virtual, como se muestra en la Ilustración 35, en la cual se puede ver como ejemplo la creación de una imagen de disco virtual de 10 GiB. La especificación de este parámetro fue adaptado a cada servidor virtual según la capacidad de almacenamiento estimada que necesitaría para alojar el servicio web que le sería correspondido.

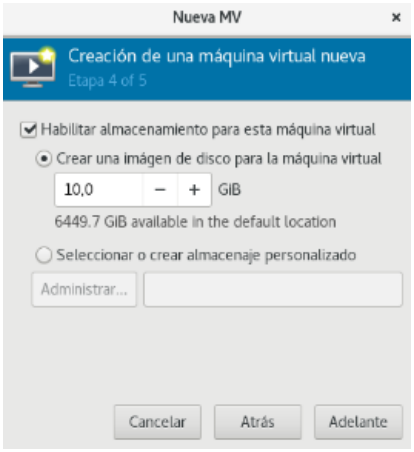


Ilustración 35: Creación de una máquina virtual de KVM - Etapa 4

Quinto paso: Especificación del nombre y la red de la máquina virtual

El último paso consistió en especificar un nombre para la nueva máquina virtual y la red en la que se comunicaría, como se indica en la Ilustración 36, en la cual se puede ver como ejemplo la asignación del nombre "NOMBRE_MV" y la red NAT por defecto denominada *default*. La especificación del nombre se adaptó a cada servidor virtual según el servicio web que alojaría y la red especificada fue la misma para todos ellos, de tipo NAT.

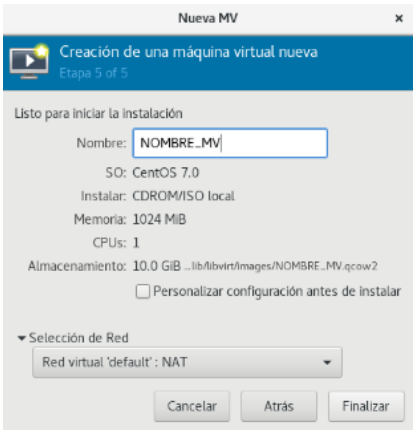


Ilustración 36: Creación de una máquina virtual de KVM - Etapa 5

Tras finalizar la creación de la máquina virtual esta se inició automáticamente dando paso así a la instalación, necesariamente mediante interfaz gráfica, de su sistema operativo.

Una vez creadas todas las máquinas virtuales requeridas fue necesario enmascarar sus direcciones IP virtuales de forma que pudieran acceder a Internet. Para ello, fue necesario añadir varias reglas en el cortafuegos tanto la tabla *filter* como de la tabla *nat* de forma que el fichero de configuración */etc/sysconfig/iptables* tuviera el siguiente contenido:

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [30:3059]
-A INPUT -i lo -m state --state NEW -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -m state --state NEW -j ACCEPT
-A INPUT -p udp -m udp -m multiport --dports 80,443 -m state --state NEW -j ACCEPT
-A INPUT -p tcp -m tcp -m multiport --dports 22,80,443 -m state --state NEW -j ACCEPT
-A INPUT -i em2 -p tcp -m tcp --dport 3389 -m state --state NEW -j ACCEPT
-A INPUT -i virbr0 -p udp -m udp -m multiport --dports 53,67 -j ACCEPT
-A INPUT -i virbr0 -p tcp -m tcp -m multiport --dports 53,67 -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i virbr0 -o virbr0 -j ACCEPT
-A FORWARD -s DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED -i virbr0 -j ACCEPT
-A FORWARD -d DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED -o virbr0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
COMMIT
*nat
:PREROUTING ACCEPT [21:1500]
:INPUT ACCEPT [18:1086]
:OUTPUT ACCEPT [2:120]
:POSTROUTING ACCEPT [2:120]
-A POSTROUTING -s DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED ! -d
DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED -p udp -m udp -j MASQUERADE --to-ports 1024-65535
-A POSTROUTING -s DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED ! -d
DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED -p tcp -m tcp -j MASQUERADE --to-ports 1024-65535
-A POSTROUTING -s DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED ! -d
DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED -j MASQUERADE
COMMIT
…
~
"/etc/sysconfig/iptables"
```

Una vez guardados los cambios se reinició el servicio de *iptables*:

```
# systemctl restart iptables
```

ANEXO X: PRIMERA CONFIGURACIÓN DEL SERVIDOR WEB VIRTUAL DE ALOJAMIENTO DEL SERVICIO WEB DE MOODLE

Modificación del nombre de host

En primer lugar, se modificó el nombre de *host* del servidor virtual de Moodle:

```
# hostnamectl set-hostname NOMBRE_HOST.ciber.ulpgc.es
```

Modificación de la configuración de red

A continuación, se configuró la interfaz de red del servidor virtual, de forma que tuviera asignada una dirección IP estática conocida para su fácil identificación. Para ello se especificaron en el fichero */etc/sysconfig/network-scripts/ifcfg-eth0* del servidor virtual los siguientes parámetros:

```
…
BOOTPROTO=none
…
ONBOOT=yes
ZONE=public
IPADDR=DIRECCIÓN_IP_SERVIDOR_VIRTUAL
PREFIX=24
GATEWAY=DIRECCIÓN_IP_SERVIDOR_HIPERVISOR
DNS1=DIRECCIÓN_IP_SERVIDOR_HIPERVISOR
~
"/etc/sysconfig/network-scripts/ifcfg-eth0"
```

Una vez guardados los cambios, se reinició el servidor virtual:

```
# reboot
```

Actualización del sistema

Se actualizaron todos los paquetes del sistema recién instalado en cada servidor virtual:

```
# yum update -y
```

Instalación del repositorio EPEL

Se requirió instalar el repositorio EPEL como prerrequisito para las subsiguientes instalaciones en el servidor:

```
# yum install epel-release -y
```

Instalación del paquete yum-utils

Se instaló el paquete *yum-utils* para la posterior gestión de los repositorios de *yum*:

```
# yum install yum-utils -y
```

Instalación del paquete policycoreutils-python

Se instaló el paquete *policycoreutils-python* para la posterior gestión de los contextos de SELinux:

```
# yum install policycoreutils-python -y
```

Instalación de wget

También se requirió instalar la herramienta *wget*:

```
# yum install wget -y
```

Habilitación de servicio mediante FirewallD: Servicio HTTP

Para permitir las conexiones entrantes al servidor web primero fue necesario habilitar, mediante una regla, el protocolo HTTP en el cortafuegos del sistema a través de *FirewallD*:

```
# firewall-cmd --zone=public --add-service=http --permanent
# firewall-cmd --reload
```

Instalación de Apache

Se instaló el servidor web de Apache:

```
# yum install httpd -y
```

Y se inició su servicio:

```
# systemctl start httpd
```

Instalación de MariaDB

Se instaló el sistema de bases de datos de MariaDB:

```
# yum install mariadb-server -y
```

Se inició su servicio:

```
# systemctl start mariadb
```

Y se procedió a realizar una instalación segura de dicho sistema:

```
# /usr/bin/mysql_secure_installation
```

Creación de la base de datos de Moodle

Una vez instalado el sistema de gestión de bases de datos, fue necesario crear una nueva base de datos para Moodle.

Para ello primero se accedió a la interfaz de MariaDB:

```
# mysql -u NOMBRE_USUARIO_MYSQL -p
```

Una vez dentro, se creó la base de datos de la siguiente manera:

```
MariaDB [(none)]> CREATE DATABASE moodle DEFAULT CHARACTER SET UTF8MB4 COLLATE utf8mb4_unicode_ci;
```

Seguidamente se creó un usuario específico para dicha base de datos:

```
MariaDB [(none)]> CREATE USER 'NOMBRE_USUARIO_BD_MOODLE'@'localhost' IDENTIFIED BY 'NUEVA_CONTRASEÑA_USUARIO_BD_MOODLE';
```

Se le otorgaron al usuario todos los privilegios sobre la base de datos:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON moodle.* TO 'NOMBRE_USUARIO_BD_MOODLE'@'localhost' IDENTIFIED BY 'CONTRASEÑA_USUARIO_BD_MOODLE' WITH GRANT OPTION;
```

Se recargó la tabla de privilegios del sistema de base de datos:

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

Y se cerró la sesión de la terminal de MariaDB:

```
MariaDB [(none)]> EXIT;
```

Instalación de PHP

Debido a que el servicio web de Moodle que estaba ejecutándose en el servidor MD era de una versión bastante desactualizada fue necesario instalar la versión 5.6 de PHP para que el servicio funcionase correctamente tras la migración.

Para ello, primero se activó el repositorio Remi correspondiente a la versión 5.6 de PHP mediante la herramienta *yum-config-manager* del paquete *yum-utils*:

```
# yum-config-manager --enable remi-php56
```

Seguidamente, se procedió a la instalación de PHP:

```
# yum install php -y
```

Y se comprobó la versión instalada:

```
# php -v
```

ANEXO XI: MIGRACIÓN DEL SERVICIO WEB DE MOODLE A SU SERVIDOR VIRTUAL CORRESPONDIENTE

Deshabilitación del servicio web en el servidor original

Para migrar este servicio web antes fue necesario asegurarse de que fuera inaccesible.

Para ello y en primer lugar, se detuvo tanto el servidor web Apache como el sistema de gestión de bases de datos MySQL en el servidor MD:

```
$ sudo service apache2 stop
$ sudo service mysql stop
```

Migración del directorio 'moodle'

Para copiar de forma remota el directorio contenedor de los documentos HTML del servicio de Moodle del servidor MD a su correspondiente servidor virtual se ejecutó en este último el siguiente comando:

```
# scp -r
NOMBRE_USUARIO@DIRECCIÓN_IP_SERVIDOR_MD:/var/www/moodle/
/var/www/html/
```

A continuación, se ajustaron los permisos de todos los ficheros de dicho directorio:

```
# chown -R root:root /var/www/html/moodle/
# chmod -R 0755 /var/www/html/moodle/
# find /var/www/html/moodle -type f -exec chmod 0644 {} \;
```

También fue necesario ajustar el contexto SELinux del directorio:

```
# semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/moodle(/.*)?'
# restorecon -Rv '/var/www/html/moodle/'
```

Así mismo, fue necesario modificar el fichero de configuración de la instancia de Moodle */var/www/html/moodle/config.php* para adaptarla al servidor virtual.

Además, fue necesario modificar el fichero */etc/httpd/conf/httpd.conf* para cambiar el valor del parámetro '*DocumentRoot'*, que especifica el directorio raíz de alojamiento del servidor web:

```
…
DocumentRoot "/var/www/html/moodle/"
…
~
"/etc/httpd/conf/httpd.conf"
```

Por último, se reinició el servidor web:

```
# systemctl restart httpd
```

Migración del directorio 'moodledata'

Por otro lado, también fue necesario copiar de forma remota el directorio contenedor de los datos y la estructura de la instancia de Moodle del servidor MD a su correspondiente servidor virtual. Para ello, se ejecutó en el servidor virtual el siguiente comando:

```
# scp -r NOMBRE_USUARIO@DIRECCIÓN_IP_SERVIDOR_MD:/var/moodledata/ /var/www/
```

Una vez realizada la transferencia, se ajustaron los permisos de todos los ficheros de dicho directorio:

```
# chown -R apache:apache /var/www/moodledata/
# chmod -R 0700 /var/www/moodledata/
# find /var/www/moodledata -type f -exec chmod 0600 {} \;
```

Por último, fue necesario ajustar el contexto SELinux del directorio:

```
# semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/moodledata(/.*)?'
# restorecon -Rv '/var/www/moodledata/'
```

Migración de los datos de las bases de datos del servicio web de Moodle

Para exportar la base de datos asociada al servicio de Moodle se ejecutó el siguiente comando en el servidor MD:

```
$ mysqldump -u NOMBRE_USUARIO_BD -p moodle > /root/moodle.sql
```

De esta forma, se guardaron los datos dicha base de datos en un fichero SQL creado, en este caso, en el directorio */root/*.

Tras esto, se copió de forma remota el fichero SQL generado al servidor virtual correspondiente, ejecutando en este último el siguiente comando:

```
# scp -r NOMBRE_USUARIO@DIRECCIÓN_IP_SERVIDOR_MD:/root/moodle.sql /root/
```

Finalmente, se importó en la base de datos creada en el servidor virtual los datos del fichero SQL ejecutando el siguiente comando:

```
# mysql -u NOMBRE_USUARIO_BD_MOODLE -p moodle < /root/moodle.sql
```

Migración de los nombres de dominio del servicio web de Moodle mediante ULPnet

Por último, para permitir el fácil acceso al servicio web de Moodle alojado en su respectivo servidor virtual fue necesario desasociar sus nombres de dominio del servidor MD y asociarlos al servidor MR a través de la herramienta de gestión de redes de ULPnet.

Configuración del proxy inverso: Habilitación del acceso al servicio web de Moodle

Como último paso en el despliegue del servicio de Moodle en un servidor virtual fue necesario permitir el acceso de los clientes a dicho servidor mediante el *proxy* inverso de NGINX instalado en el servidor hipervisor.

Para ello, fue necesario crear en el directorio */etc/nginx/conf.d/* del servidor hipervisor un nuevo fichero de configuración llamado, en este caso, *moodle.conf* con la siguiente estructura:

```
server {
        listen 80;
        server_name NOMBRE(S)_DOMINIO_SERVICIO_WEB_MOODLE;

        location / {
                access_log off;
        proxy_pass http://DIRECCIÓN_IP_SERVIDOR_WEB_VIRTUAL_MOODLE;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
         }
}
~
"/etc/nginx/conf.d/moodle.conf"
```

De esta manera, toda solicitud HTTP realizada utilizando cualquiera de los nombres de dominio especificados tras la directiva *server_name* será transmitida por el servidor hipervisor, en su función como *proxy* inverso, al servidor virtual web que aloja el servicio de Moodle: *http://**DIRECCIÓN_IP_SERVIDOR_WEB_VIRTUAL_MOODLE***. Una vez el servidor virtual haya procesado la solicitud, el servidor hipervisor rescatará la respuesta generada y la enviará de vuelta al cliente de origen.

Para transmitir dichas solicitudes al servidor web virtual correspondiente fue necesario emplear la directiva *proxy_pass* dentro del bloque *location* [43].

Una vez creado el fichero, se recargó el sistema de NGINX del servidor hipervisor:

```
# nginx -s reload
```

ANEXO XII: PRIMERA ACTUALIZACIÓN DEL SERVICIO WEB DE MOODLE

Instalación de Git

En primer lugar, se instaló la herramienta Git para facilitar la actualización de la instancia de Moodle:

```
# yum install git -y
```

Detención de XAMPP

A continuación, se detuvo la distribución de XAMPP en el servidor virtual dejando de esta forma el servicio web de Moodle fuera de línea:

```
# /opt/lampp/lampp stop
```

Copia de seguridad de los directorios 'moodle' y 'moodledata'

Se realizó una copia de seguridad de los directorios */opt/lampp/htdocs/moodle/* y */opt/lampp/moodledata/* del servidor virtual la cual se guardó en un disco duro externo provisto por el instituto.

Copia de seguridad de la base de datos de Moodle

Se realizó una copia de seguridad de la base de datos del servicio web de Moodle del servidor virtual:

```
# /opt/lampp/bin/mysqldump -u NOMBRE_USUARIO_BD -p moodle > /root/moodle.sql
```

Dicha copia de seguridad se guardó en un disco duro externo provisto por el instituto.

Inicio de XAMPP

Una vez realizadas las copias de seguridad, se volvió a iniciar la distribución de XAMPP en el servidor virtual:

```
# /opt/lampp/lampp start
```

Habilitación del modo de mantenimiento de Moodle

A continuación, fue necesario acceder al sitio web de Moodle como administrador y habilitar el modo de mantenimiento en *Administración* > *Administración del sitio* > *Servidor* > *Modo de mantenimiento* para poder proceder correctamente con la actualización de la instancia.

Preparación de la actualización de Moodle mediante Git

Siguiendo la guía de administración de Git ofrecida por Moodle [44], se realizaron algunas configuraciones mediante Git para proceder a la actualización.

En primer lugar, se accedió al directorio */var/www/html/:*

```
# cd /var/www/html/
```

Se inicializó un nuevo repositorio local como clonación del repositorio remoto *moodle.git* de Moodle:

```
# git clone git://git.moodle.org/moodle.git
```

De esta manera, se creó en el directorio */var/www/html/* un nuevo directorio */moodle/* en el cual se descargaron los ficheros correspondientes a todas las versiones disponibles en ese momento de Moodle.

Seguidamente, se accedió al directorio recién creado */moodle/*:

```
# cd moodle/
```

Se listaron todas las ramas disponibles de los repositorios (local y remoto):

```
# git branch -a
```

Se creó una nueva rama en el repositorio local llamada, en este caso, MOODLE_31_STABLE configurada para rastrear la rama MOODLE_31_STABLE del repositorio remoto:

```
# git branch --track MOODLE_31_STABLE origin/MOODLE_31_STABLE
```

Por último, se cambió la rama activa a la rama local recién creada:

```
# git checkout MOODLE_31_STABLE
```

Como resultado de estas configuraciones, el directorio */var/www/html/moodle/* pasó a contener todos los ficheros necesarios para la actualización de Moodle, en este caso, a la versión 3.1, facilitando además el proceso, mediante Git, de futuras actualizaciones.

Inicio de la actualización de Moodle

Una vez actualizado el directorio */var/www/html/moodle/* con los ficheros correspondientes a la versión 3.1 de Moodle se decidió proceder con la actualización de la instancia mediante interfaz gráfica.

Para ello se accedió desde el navegador web del servidor hipervisor al servicio web de Moodle mediante la dirección IP de su servidor virtual correspondiente.

En la primera pantalla, se configuró nuevamente la nueva instancia de Moodle y se copiaron esos datos al fichero */var/www/html/moodle/config.php* del servidor virtual.

Tras esto, se presentó otra pantalla con la comprobación de compatibilidad y actualización de las extensiones instaladas en ese momento en la instancia de Moodle.

Además, presentó una serie de configuraciones recomendables antes de proceder con la actualización para evitar un mal funcionamiento de la instancia tras esta.

Recomendación de actualización de Moodle: Reconversión del formato de archivo de las tablas de la base de datos y cambio del conjunto de caracteres

Antes de la actualización se reconvirtió el formato de archivo de todas las tablas de la base de datos de Moodle de "Antelope" a "Barracuda" y, además, se cambió el conjunto de caracteres de "utf8" a "utf8mb4".

En primer lugar, se modificó el fichero */etc/my.cnf.d/client.cnf* y bajo la etiqueta *[client]* se añadió la siguiente línea:

```
…
[client]
default-character-set = utf8mb4
…
~
"/etc/my.cnf.d/client.cnf"
```

Se modificó el fichero */etc/my.cnf.d/server.conf* y bajo la etiqueta *[mysqld]* se añadieron las siguientes líneas:

```
…
[mysqld]
innodb_file_format = Barracuda
innodb_file_per_table = 1
innodb_large_prefix = 1

character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci
skip-character-set-client-handshake
…
~
"/etc/my.cnf.d/server.conf"
```

Se modificó el fichero */etc/my.cnf.d/mysql-clients.cnf* y bajo la etiqueta *[mysql]* se añadió la siguiente línea:

```
…
[mysql]
default-character-set = utf8mb4
…
~
"/etc/my.cnf.d/mysql-clients.cnf"
```

Se accedió al directorio */var/www/html/moodle/*:

```
# cd /var/www/html/moodle/
```

Se listaron las tablas de la base de datos de Moodle, tanto las comprimidas como las que no:

```
# php admin/cli/mysql_compressed_rows.php --list
```

De esta lista se hizo anotación de aquellas tablas que no estaban comprimidas.

A continuación, se accedió a la interfaz de línea de comandos de MariaDB:

```
# mysql -u root -p
```

Una vez dentro, se accedió a la base de datos de Moodle:

```
MariaDB [(none)]> USE moodle;
```

Se cambió el modo SQL a estricto para todas las tablas, de forma que los posibles fallos en la modificación de cada tabla solo afectaran a la primera fila de cada una y no se propagara hacia las demás:

```
MariaDB [(none)]> SET SESSION sql_mode=STRICT_ALL_TABLES;
```

Se cambió el formato de archivo de las tablas del motor INNODB al formato “Barracuda”:

```
MariaDB [(none)]> SET GLOBAL innodb_file_format=Barracuda;
```

Se comprimió cada tabla no comprimida de las que se hizo anotación anteriormente:

```
MariaDB [(none)]> ALTER TABLE NOMBRE_TABLA
ROW_FORMAT=Compressed;
```

Se cerró sesión en la interfaz de MariaDB:

```
MariaDB [(none)]> EXIT;
```

Se cambió la colación de la base de datos de Moodle:

```
# php admin/cli/mysql_collation.php --collation=utf8mb4_unicode_ci
```

Por último, se comprobó que la variable *$CFG->dboptions* del fichero de configuración de Moodle utilizara la misma colación para conectarse a la base de datos:

```
…
$CFG->dboptions = array(
…
‘dbcollation’ => ‘utf8mb4_unicode_ci’,
…
);
…
~
“/var/www/html/moodle/config.php”
```

Recomendación de actualización de Moodle: Instalación del módulo de PHP “xmlrpc”

Para instalar el módulo “xmlrp” se ejecutó en el servidor virtual el siguiente comando:

```
# yum install php-xmlrpc -y
```

Recomendación de actualización de Moodle: Instalación del módulo de PHP “soap”

Para instalar el módulo “soap” se ejecutó en el servidor virtual el siguiente comando:

```
# yum install php-soap -y
```

Recomendación de actualización de Moodle: Instalación del módulo de PHP “intl”

Para instalar el módulo “intl” se ejecutó en el servidor virtual el siguiente comando:

```
# yum install php-intl -y
```

Recomendación de actualización de Moodle: Instalación del módulo de PHP “opcache”

Para instalar el módulo “opcache” se ejecutó en el servidor virtual el siguiente comando:

```
# yum install php-opcache -y
```

Finalización de la actualización de Moodle

Finalmente, se reiniciaron tanto el servidor web como el sistema de bases de datos del servidor virtual:

```
# systemctl restart httpd && systemctl restart mariadb
```

Una vez realizadas todas las acciones recomendadas se procedió con la finalización de la actualización y se deshabilitó el modo mantenimiento de Moodle.

ANEXO XIII: SEGUNDA CONFIGURACIÓN DEL SERVIDOR WEB VIRTUAL DE ALOJAMIENTO DEL SERVICIO WEB DE MOODLE

Instalación de XAMPP

Se obtuvo, mediante la herramienta *wget*, el instalador de la última versión de XAMPP de ese momento:

```
# wget RUTA_WEB_INSTALADOR_XAMPP
```

Una vez descargado el fichero, se le otorgó el permiso de ejecución:

```
# chmod +x FICHERO_INSTALADOR_XAMPP
```

Y, por último, se ejecutó el instalador:

```
# ./FICHERO_INSTALADOR_XAMPP
```

Una vez finalizada la instalación de la distribución se procedió a su configuración.

Modificación del fichero /opt/lampp/etc/extra/httpd-xampp.conf

Se modificó el fichero */opt/lampp/etc/extra/httpd-xampp.conf* para permitir a todos el acceso a la distribución de XAMPP.

Para ello, fue necesario comentar en dicho archivo la línea '*Require local'* y añadir la línea '*Require all granted'*:

```
…
# Require local
Require all granted
~
"/opt/lampp/etc/extra/httpd-xampp.conf"
```

Modificación del fichero /opt/lampp/etc/php.ini

Se modificó el fichero */opt/lampp/etc/php.ini* para cambiar el valor por defecto de algunos parámetros de PHP.

En este caso, se cambió el valor de los parámetros *post_max_size* y *upload_max_filesize* a 1024 MB para poder subir ficheros (generalmente de tipo SQL) con un tamaño máximo de 1GB al sistema de bases de datos del servidor virtual de Moodle:

```
…
post_max_size=1024M
…
upload_max_filesize=1024M
~
“/opt/lampp/etc/php.ini”
```

Modificación del fichero /opt/lampp/phpmyadmin/config.inc.php

Se modificó el fichero */opt/lampp/phpmyadmin/config.inc.php* para cambiar el valor por defecto de algunos parámetros de phpMyAdmin.

En primer lugar, y con el fin de obtener mayor seguridad en el acceso a phpMyAdmin se estableció el método de autenticación basado en *cookies*. Para ello, se cambió el valor del parámetro *$cfg['Servers'][$i]['auth_type']* (por defecto, ‘*config*’) al valor ‘*cookie*’:

```
…
$cfg[‘Servers’][$i][‘auth_type’] = ‘cookie’;
…
~
“/opt/lampp/phpmyadmin/config.inc.php”
```

Con el fin de obtener una mayor seguridad en la autenticación mediante nombre de usuario y contraseña se cambió el valor del parámetro *$cfg[‘blowfish_secret’]* (por defecto, ‘*xampp*’) sustituyéndolo por una clave simétrica generada aleatoriamente mediante el método de cifrado *Blowfish*:

```
…
$cfg[‘blowfish_secret’] = ‘CLAVE_BLOWFISH’;
…
~
“/opt/lampp/phpmyadmin/config.inc.php”
```

Con el fin de obtener una mayor seguridad en la autenticación del usuario *root* se cambió el valor del parámetro *$cfg['Servers'][$i]['password']* (por defecto con un valor vacío) añadiendo la contraseña deseada para dicho usuario:

```
…
$cfg['Servers'][$i]['password'] = ‘CONTRASEÑA_ROOT’;
…
~
“/opt/lampp/phpmyadmin/config.inc.php”
```

Con el fin de obtener una mayor seguridad en la autenticación del usuario *pma* se cambió el valor del parámetro *$cfg['Servers'][$i]['controlpass']* (por defecto con un valor vacío) añadiendo la contraseña deseada para dicho usuario:

```
…
$cfg['Servers'][$i]['controlpass'] = ‘CONTRASEÑA_PMA’;
…
~
“/opt/lampp/phpmyadmin/config.inc.php”
```

Por último y con el fin de habilitar el almacenamiento de archivos temporales del servidor web se añadió el parámetro *$cfg[‘TempDir’]* con el valor *‘/tmp*’:

```
…
$cfg['TempDir'] = ‘/tmp’;
…
~
“/opt/lampp/phpmyadmin/config.inc.php”
```

Automatización del inicio de XAMPP

Para facilitar el inicio de todos los componentes de la distribución de XAMPP se decidió automatizarlo de forma que este se ejecutara justo después del arranque del servidor.

Para ello, primero se creó un enlace simbólico del *script* de ejecución de XAMPP en el directorio */etc/init.d/* :

```
# ln -s /opt/lampp/lampp /etc/init.d/lampp
```

Y a continuación, se añadió el *script* a la gestión de la herramienta *chkconfig*, que se encargaría de iniciar el *script* con cada arranque del servidor virtual de Moodle:

```
# chkconfig --add lampp
```

Exportación de los datos de la base de datos de Moodle de MariaDB

Se exportaron los datos de la base de datos del servicio web de Moodle a un fichero SQL:

```
# mysqldump -u NOMBRE_USUARIO_BD -p moodle > /root/moodle.sql
```

Detención de Apache y MariaDB

Se detuvo tanto el servidor web *httpd* como el sistema de bases de datos de MariaDB:

```
# systemctl stop httpd && systemctl stop mariadb
```

Inicio de XAMPP

A continuación, se inició la distribución de XAMPP:

```
# /opt/lampp/lampp start
```

Modificación de las tablas de privilegios MySQL mediante phpMyAdmin

Una vez iniciada la distribución de XAMPP, fue necesario establecer las mismas contraseñas especificadas en el fichero */opt/lampp/phpmyadmin/config.inc.php* en las tablas de privilegios MySQL de los usuarios *root* y *pma*.

Comprobación de seguridad de XAMPP

Se mejoró la seguridad de la distribución de XAMPP:

```
# /opt/lampp/lampp security
```

Modificación del fichero /opt/lampp/etc/my.cnf

Se incluyeron en el fichero de configuración del sistema de bases de datos de XAMPP las siguientes líneas:

```
…
[client]
…
default-character-set = utf8mb4
…
[mysqld]
…
innodb_file_format = Barracuda
innodb_file_per_table = 1
innodb_large_prefix = 1

character-set-server = utf8mb4
collation-server = utf8mb4_unicode_ci
skip-character-set-client-handshake
…
[mysql]
…
default-character-set = utf8mb4
…
~
"/opt/lampp/etc/my.cnf"
```

Creación de la base de datos de Moodle en XAMPP

Una vez instalado el sistema de gestión de bases de datos, fue necesario crear una nueva base de datos para Moodle.

Para ello primero se accedió a la interfaz de MariaDB:

```
# mysql -u NOMBRE_USUARIO_MYSQL -p
```

Una vez dentro, se creó la base de datos de la siguiente manera:

```
MariaDB [(none)]> CREATE DATABASE moodle DEFAULT CHARACTER SET UTF8MB4 COLLATE utf8mb4_unicode_ci;
```

Seguidamente se creó un usuario específico para dicha base de datos:

```
MariaDB [(none)]> CREATE USER 'NOMBRE_USUARIO_BD_MOODLE'@'localhost' IDENTIFIED BY 'NUEVA_CONTRASEÑA_USUARIO_BD_MOODLE';
```

Se le otorgaron al usuario todos los privilegios sobre la base de datos:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON moodle.* TO 'NOMBRE_USUARIO_BD_MOODLE'@'localhost' IDENTIFIED BY 'CONTRASEÑA_USUARIO_BD_MOODLE' WITH GRANT OPTION;
```

Se recargó la tabla de privilegios del sistema de bases de datos:

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

Y se cerró la sesión de la terminal de MariaDB:

```
MariaDB [(none)]> EXIT;
```

Importación de los datos de la base de datos de Moodle a la base de datos de XAMPP

Se importaron los datos del fichero SQL previamente generado a la base de datos de Moodle creada en la distribución de XAMPP:

```
# /opt/lampp/bin/mysql -u NOMBRE_USUARIO_BD -p moodle < /root/moodle.sql
```

Transferencia del directorio 'moodle' a XAMPP

Se movió el directorio */var/www/html/moodle/* a la distribución de XAMPP:

```
# mv /var/www/html/moodle/ /opt/lampp/htdocs/
```

Transferencia del directorio 'moodledata' a XAMPP

Se movió el directorio */var/www/moodledata/* a la distribución de XAMPP:

```
# mv /var/www/moodledata/ /opt/lampp/
```

Además, se aseguró el directorio con el nuevo propietario:

```
# chown -R daemon:daemon /opt/lampp/moodledata/
```

Modificación del fichero /opt/lampp/htdocs/moodle/config.php

Fue necesario modificar el fichero */opt/lampp/htdocs/moodle/config.php* para declarar la nueva localización del fichero de datos de Moodle:

```
…
$CFG->dataroot   =   '/opt/lampp/moodledata';
…
~
"/opt/lampp/htdocs/moodle/config.php"
```

Modificación del fichero /opt/lampp/etc/httpd.conf

Se requirió modificar el fichero */opt/lampp/etc/httpd.conf* para cambiar el valor del parámetro '*DocumentRoot'*, que especifica el directorio raíz de alojamiento del servidor web:

```
…
DocumentRoot "/opt/lampp/htdocs/moodle/"
…
~
"/opt/lampp/etc/httpd.conf"
```

Reinicio de XAMPP

Por último, se reinició la distribución de XAMPP:

```
# /opt/lampp/lampp restart
```

Desinstalación de Apache

Se desinstaló el servidor web de Apache:

```
# yum remove httpd -y
```

Desinstalación de MariaDB

Se desinstaló el sistema de gestión de bases de datos de MariaDB:

```
# yum remove mariadb-server -y
```

ANEXO XIV: SEGUNDA ACTUALIZACIÓN DEL SERVICIO WEB DE MOODLE

Detención de XAMPP

A continuación, se detuvo la distribución de XAMPP en el servidor virtual dejando de esta forma el servicio web de Moodle fuera de línea:

```
# /opt/lampp/lampp stop
```

Copia de seguridad de los directorios 'moodle' y 'moodledata'

Se realizó una copia de seguridad de los directorios */opt/lampp/htdocs/moodle/* y */opt/lampp/moodledata/* del servidor virtual la cual se guardó en un disco duro externo provisto por el instituto.

Copia de seguridad de la base de datos de Moodle

Se realizó una copia de seguridad de la base de datos del servicio web de Moodle del servidor virtual:

```
# /opt/lampp/bin/mysqldump -u NOMBRE_USUARIO_BD -p moodle >
/root/moodle.sql
```

Dicha copia de seguridad se guardó en un disco duro externo provisto por el instituto.

Inicio de XAMPP

Una vez realizadas las copias de seguridad, se volvió a iniciar la distribución de XAMPP en el servidor virtual:

```
# /opt/lampp/lampp start
```

Habilitación del modo de mantenimiento de Moodle

A continuación, fue necesario acceder al sitio web de Moodle como administrador y habilitar el modo de mantenimiento en *Administración* > *Administración del sitio* > *Servidor* > *Modo de mantenimiento* para poder proceder correctamente con la actualización de la instancia.

Preparación de la actualización de Moodle mediante Git

A continuación, se realizaron algunas configuraciones mediante Git para proceder a la actualización.

En primer lugar, se accedió al directorio */opt/lampp/htdocs/moodle/*:

```
# cd /opt/lampp/htdocs/moodle/
```

Seguidamente, se listaron todas las ramas disponibles de los repositorios (local y remoto):

```
# git branch -a
```

Se creó una nueva rama en el repositorio local llamada, en este caso *MOODLE_36_STABLE*, configurada para rastrear la rama *MOODLE_36_STABLE* del repositorio remoto:

```
# git branch --track MOODLE_36_STABLE origin/MOODLE_36_STABLE
```

Se cambió la rama activa a la rama local recién creada:

```
# git checkout MOODLE_36_STABLE
```

Por último, se descargaron en el directorio */opt/lampp/htdocs/moodle/* todos los ficheros necesarios para la actualización de Moodle, en este caso, a la versión 3.6:

```
# git pull
```

Inicio de la actualización de Moodle

Una vez actualizado el directorio */opt/lampp/htdocs/moodle/* con los ficheros correspondientes a la versión 3.6 de Moodle se decidió proceder con la actualización de la instancia mediante interfaz gráfica.

En la primera pantalla, se configuró nuevamente la nueva instancia de Moodle.

Tras esto, se presentó otra pantalla con la comprobación de compatibilidad y actualización de las extensiones instaladas en ese momento en la instancia de Moodle, así como una serie de configuraciones recomendables, entre otros, antes de proceder con la actualización de la versión.

Recomendación de actualización de Moodle: Habilitación del módulo de PHP "opcache"

Se habilitó el módulo de PHP *opcache* descomentando en el fichero */opt/lampp/etc/php.ini* la siguiente línea:

```
…
zend_extension=opcache.so
…
~
"/opt/lampp/etc/php.ini"
```

Error de actualización de Moodle: Componente "es_es" no encontrado

Se mostró un error sobre un componente no encontrado, en concreto el correspondiente al lenguaje "es_es". Para resolver este error simplemente se eliminó el fichero */opt/lampp/moodledata/lang/es_es* del servidor virtual, pues estaba obsoleto:

```
# rm -rf /opt/lampp/moodledata/lang/es_es
```

Finalización de la actualización de Moodle

Una vez realizadas todas las acciones recomendadas se procedió con la finalización de la actualización y se deshabilitó el modo mantenimiento de Moodle.

Reinstalación de módulo de Moodle: Questionnaire

Además, fue necesario instalar una nueva versión del módulo de Moodle "questionnaire" que en ese momento estaba obsoleto.

Para ello, en primer lugar, se descargó el módulo en forma de fichero ZIP:

```
# wget -P /root/ RUTA_WEB_MÓDULO_MOODLE
```

Se descomprimió dicho fichero y se movió el directorio resultante al directorio de módulos de actividad de Moodle:

```
# unzip /root/FICHERO_ZIP_MÓDULO_MOODLE.zip
# mv /root/DIRECTORIO_MÓDULO_MOODLE/ /opt/lampp/htdocs/moodle/mod/
```

Además, se aseguró el directorio y su contenido con los permisos adecuados:

```
# chmod -R 0755
/opt/lampp/htdocs/moodle/mod/DIRECTORIO_MÓDULO_MOODLE/
```

```
# find /opt/lampp/htdocs/moodle/mod/DIRECTORIO_MÓDULO_MOODLE/ -type f -
exec chmod 0644 {} \;
```

Finalmente, se accedió como administrador a la página de gestión de módulos de la instancia de Moodle y se instaló el módulo añadido.

Programación de tareas de Moodle mediante Cron

Adicionalmente, se requirió la programación de la ejecución de forma regular y en segundo plano de las tareas y actividades de Moodle a diferentes intervalos agendados mediante Cron. Para ello se modificó el fichero *crontab* del usuario *daemon*:

```
# crontab -u daemon -e
```

Añadiendo, en este caso, la siguiente línea:

```
* * * * * /opt/lampp/bin/php /opt/lampp/htdocs/moodle/admin/cli/cron.php > /dev/null
```

Por último, se volvió a activar el repositorio Remi correspondiente a la versión 7.2 de PHP:

```
# yum-config-manager --enable remi-php72
```

ANEXO XV: TERCERA ACTUALIZACIÓN DEL SERVICIO WEB DE MOODLE

Detención de XAMPP

A continuación, se detuvo la distribución de XAMPP en el servidor virtual dejando de esta forma el servicio web de Moodle fuera de línea:

```
# /opt/lampp/lampp stop
```

Copia de seguridad de los directorios 'moodle' y 'moodledata'

Se realizó una copia de seguridad de los directorios */opt/lampp/htdocs/moodle/* y */opt/lampp/moodledata/* del servidor virtual la cual se guardó en un disco duro externo provisto por el instituto.

Copia de seguridad de la base de datos de Moodle

Se realizó una copia de seguridad de la base de datos del servicio web de Moodle del servidor virtual:

```
# /opt/lampp/bin/mysqldump -u NOMBRE_USUARIO_BD -p moodle >
/root/moodle.sql
```

Dicha copia de seguridad se guardó en un disco duro externo provisto por el instituto.

Inicio de XAMPP

Una vez realizadas las copias de seguridad, se volvió a iniciar la distribución de XAMPP en el servidor virtual:

```
# /opt/lampp/lampp start
```

Habilitación del modo de mantenimiento de Moodle

A continuación, fue necesario acceder al sitio web de Moodle como administrador y habilitar el modo de mantenimiento en *Administración* > *Administración del sitio* > *Servidor* > *Modo de mantenimiento* para poder proceder correctamente con la actualización de la instancia.

Preparación de la actualización de Moodle mediante Git

A continuación, se realizaron algunas configuraciones mediante Git para proceder a la actualización.

Se accedió al directorio */opt/lampp/htdocs/moodle/*:

```
# cd /opt/lampp/htdocs/moodle/
```

Seguidamente, se listaron todas las ramas disponibles de los repositorios (local y remoto):

```
# git branch -a
```

Se creó una nueva rama en el repositorio local llamada, en este caso, MOODLE_37_STABLE configurada para rastrear la rama MOODLE_36_STABLE del repositorio remoto:

```
# git branch --track MOODLE_37_STABLE origin/MOODLE_36_STABLE
```

Se cambió la rama activa a la rama local recién creada:

```
# git checkout MOODLE_37_STABLE
```

Por último, se descargaron en el directorio */opt/lampp/htdocs/moodle/* todos los ficheros necesarios para la actualización de Moodle, en este caso, a la versión 3.7:

```
# git pull
```

Actualización de Moodle

Una vez actualizado el directorio */opt/lampp/htdocs/moodle/* con los ficheros correspondientes a la versión 3.7 de Moodle se decidió proceder con la actualización de la instancia mediante interfaz gráfica.

Una vez realizada la configuración y la actualización de la instancia y sus componentes, se deshabilitó el modo mantenimiento de Moodle.

ANEXO XVI: CONFIGURACIÓN DEL RESTO DE SERVIDORES WEB VIRTUALES

Modificación del nombre de host

En primer lugar, se modificó el nombre de *host* de cada servidor virtual:

```
# hostnamectl set-hostname NOMBRE_HOST.ciber.ulpgc.es
```

Modificación de la configuración de red

A continuación, se configuró la interfaz de red de cada servidor virtual, de forma que cada uno de ellos tuviera asignada una dirección IP estática conocida para su fácil identificación. Para ello se especificaron en el fichero */etc/sysconfig/network-scripts/ifcfg-eth0* de cada servidor virtual los siguientes parámetros:

```
…
BOOTPROTO=none
…
ONBOOT=yes
ZONE=public
IPADDR=DIRECCIÓN_IP_SERVIDOR_VIRTUAL
PREFIX=24
GATEWAY=DIRECCIÓN_IP_SERVIDOR_HIPERVISOR
DNS1=DIRECCIÓN_IP_SERVIDOR_HIPERVISOR
~
"/etc/sysconfig/network-scripts/ifcfg-eth0"
```

Y, una vez guardados los cambios, se reinició el servidor virtual correspondiente:

```
# reboot
```

Actualización del sistema

Se actualizaron todos los paquetes del sistema recién instalado en cada servidor virtual:

```
# yum update -y
```

Instalación de wget

Para poder descargar la distribución de XAMPP desde la interfaz de línea de comandos primero fue necesario instalar la herramienta *wget*:

```
# yum install wget -y
```

Habilitación de servicio mediante FirewallD: Servicio HTTP

Para permitir las conexiones entrantes al servidor web primero fue necesario habilitar, mediante una regla, el protocolo HTTP en el cortafuegos del sistema a través de *FirewallD*:

```
# firewall-cmd --zone=public --add-service=http --permanent
# firewall-cmd --reload
```

Instalación de XAMPP

Para cada servidor virtual se obtuvo de la página oficial de Apache Friends el instalador de la última versión de XAMPP en ese momento:

```
# wget RUTA_WEB_INSTALADOR_XAMPP
```

Una vez descargado el fichero, se le otorgó el permiso de ejecución:

```
# chmod +x FICHERO_INSTALADOR_XAMPP
```

Por último, se ejecutó el instalador:

```
# ./FICHERO_INSTALADOR_XAMPP
```

Una vez finalizada la instalación de la distribución se procedió a su configuración.

Modificación del fichero /opt/lampp/etc/extra/httpd-xampp.conf

Se modificó el fichero */opt/lampp/etc/extra/httpd-xampp.conf* para permitir a todos el acceso a la distribución de XAMPP.

Para ello, fue necesario comentar en dicho archivo la línea '*Require local'* y añadir la línea '*Require all granted'*:

```
…
# Require local
Require all granted
~
"/opt/lampp/etc/extra/httpd-xampp.conf"
```

Modificación del fichero /opt/lampp/etc/php.ini

Se modificó el fichero */opt/lampp/etc/php.ini* para cambiar el valor por defecto de algunos parámetros de PHP.

En este caso, se cambió el valor de los parámetros *post_max_size* y *upload_max_filesize* a 1024 MB para poder subir ficheros (generalmente de tipo SQL) con un tamaño máximo de 1GB al sistema de bases de datos del servidor virtual correspondiente:

```
…
post_max_size=1024M
…
upload_max_filesize=1024M
~
"/opt/lampp/etc/php.ini"
```

Modificación del fichero /opt/lampp/phpmyadmin/config.inc.php

Se modificó el fichero */opt/lampp/phpmyadmin/config.inc.php* para cambiar el valor por defecto de algunos parámetros de phpMyAdmin.

En primer lugar, y con el fin de obtener mayor seguridad en el acceso a phpMyAdmin se estableció el método de autenticación basado en *cookies*. Para ello, se cambió el valor del parámetro *$cfg['Servers'][$i]['auth_type']* (por defecto, '*config*') al valor '*cookie*':

```
…
$cfg['Servers'][$i]['auth_type'] = 'cookie';
…
~
"/opt/lampp/phpmyadmin/config.inc.php"
```

Con el fin de obtener una mayor seguridad en la autenticación mediante nombre de usuario y contraseña se cambió el valor del parámetro *$cfg['blowfish_secret']* (por defecto, '*xampp*') sustituyéndolo por una clave simétrica generada aleatoriamente mediante el método de cifrado *Blowfish*:

```
…
$cfg['blowfish_secret'] = 'CLAVE_BLOWFISH';
…
~
"/opt/lampp/phpmyadmin/config.inc.php"
```

Con el fin de obtener una mayor seguridad en la autenticación del usuario *root* se cambió el valor del parámetro *$cfg['Servers'][$i]['password']* (por defecto con un valor vacío) añadiendo la contraseña deseada para dicho usuario:

```
…
$cfg['Servers'][$i]['password'] = 'CONTRASEÑA_ROOT';
…
~
"/opt/lampp/phpmyadmin/config.inc.php"
```

Con el fin de obtener una mayor seguridad en la autenticación del usuario *pma* se cambió el valor del parámetro *$cfg['Servers'][$i]['controlpass']* (por defecto con un valor vacío) añadiendo la contraseña deseada para dicho usuario:

```
…
$cfg['Servers'][$i]['controlpass'] = '***CONTRASEÑA_PMA***';
…
~
"/opt/lampp/phpmyadmin/config.inc.php"
```

Por último y con el fin de habilitar el almacenamiento de archivos temporales del servidor web se añadió el parámetro *$cfg['TempDir']* con el valor *'/tmp*':

```
…
$cfg['TempDir'] = '/tmp';
…
~
"/opt/lampp/phpmyadmin/config.inc.php"
```

Automatización del inicio de XAMPP

Para facilitar el inicio de todos los componentes de la distribución de XAMPP se decidió automatizarlo de forma que este se ejecutara justo después del arranque del servidor virtual.

Para ello, primero se creó un enlace simbólico del *script* de ejecución de XAMPP en el directorio */etc/init.d/* :

```
# ln -s /opt/lampp/lampp /etc/init.d/lampp
```

Y a continuación, se añadió el *script* a la gestión de la herramienta *chkconfig*, que se encargaría de iniciar el *script* con cada arranque del servidor virtual:

```
# chkconfig --add lampp
```

Inicio de XAMPP

A continuación, se inició el servidor web:

```
# /opt/lampp/lampp start
```

Modificación de las tablas de privilegios MySQL mediante phpMyAdmin

Una vez iniciado el servidor web, fue necesario establecer las mismas contraseñas especificadas en el fichero */opt/lampp/phpmyadmin/config.inc.php* en las tablas de privilegios MySQL de los usuarios *root* y *pma*.

Comprobación de seguridad de XAMPP

Con el objetivo de mejorar la seguridad del servidor web se ejecutó el siguiente comando:

```
# /opt/lampp/lampp security
```

Reinicio de XAMPP

Por último, se reinició la distribución de XAMPP en el servidor virtual:

```
# /opt/lampp/lampp restart
```

ANEXO XVII: MIGRACIÓN DE CADA SERVICIO WEB A SU SERVIDOR WEB VIRTUAL CORRESPONDIENTE

Deshabilitación de un servicio web en el servidor original

Para ello y en primer lugar, se detuvo la distribución de XAMPP del servidor original:

```
# /opt/lampp/lampp stop
```

Una vez detenida la distribución, se movió el directorio contenedor de los documentos HTML del servicio web que se iba a migrar a un directorio superior:

```
# mv /opt/lampp/htdocs/DIRECTORIO_PROYECTO_WEB/ /opt/lampp/
```

Seguidamente, se volvió a iniciar la distribución de XAMPP en el servidor original:

```
#  /opt/lampp/lampp start
```

Migración de los documentos HTML de un servicio web

Para transferir de forma remota el directorio contenedor de los documentos HTML de un servicio web del servidor original a su correspondiente servidor virtual se ejecutó, en este último, el siguiente comando:

```
# scp -r
root@DIRECCIÓN_IP_SERVIDOR_ORIGINAL:/opt/lampp/DIRECTORIO_PROYECTO_WEB/ /opt/lampp/htdocs/
```

Migración de los datos de las bases de datos de un servicio web

Para exportar las bases de datos asociadas a un servicio web se ejecutó, por cada base de datos asociada, el siguiente comando en el servidor original:

```
# /opt/lampp/bin/mysqldump -u NOMBRE_USUARIO_BD -p NOMBRE_BD > NOMBRE_FICHERO_BD.sql
```

De esta forma, se guardaron los datos de cada base de datos del servicio que se iba a migrar en un fichero SQL independiente, en este caso, creado en el directorio */root/* del servidor original.

Tras esto, se transfirieron de forma remota todos los ficheros SQL generados al servidor virtual correspondiente. Para ello se ejecutó, por cada fichero generado, el siguiente comando en dicho servidor:

```
# scp
root@DIRECCIÓN_IP_SERVIDOR_ORIGINAL:/root/NOMBRE_FICHERO_BD.sql
/root/
```

Luego, fue necesario crear en el servidor virtual una base de datos por cada fichero SQL transferido del servidor original. Para ello, por cada fichero, se siguió el siguiente proceso:

1) Se accedió a la interfaz del sistema de bases de datos, en este caso MariaDB, con el siguiente comando:

   ```
   # /opt/lampp/bin/mysql -u NOMBRE_USUARIO_BD -p
   ```

2) Dentro de la interfaz, se creó la base de datos correspondiente, con el mismo nombre que la base de datos original:

   ```
   MariaDB [(none)]> CREATE DATABASE NOMBRE_BD;
   ```

3) E, inmediatamente, se finalizó la conexión con la interfaz:

   ```
   MariaDB [(none)]> exit;
   ```

Finalmente, se importaron en el servidor virtual correspondiente los datos de cada fichero SQL en su respectiva base de datos ejecutando, por cada fichero SQL, el siguiente comando:

```
# /opt/lampp/bin/mysql -u NOMBRE_USUARIO_BD -p -h localhost NOMBRE_BD <
NOMBRE_BD.sql
```

Reinicio de XAMPP

Tras migrar los documentos HTML y las bases de datos de un servicio web al servidor virtual correspondiente se reinició la distribución de XAMPP de este último:

```
# /opt/lampp/lampp restart
```

Migración de los nombres de dominio de un servicio web mediante ULPnet

Por último, para permitir el fácil acceso a un servicio web alojado en su respectivo servidor virtual fue necesario desasociar sus nombres de dominio del servidor original y asociarlos al servidor MR a través de la herramienta de gestión de redes de ULPnet.

Configuración del proxy inverso del servidor MR: Habilitación del acceso a un servicio web

Para permitir el acceso de los clientes desde Internet a un servicio web alojado en un servidor virtual se requirió configurar el *proxy* inverso instalado en el servidor MR.

Para ello, fue necesario crear en el directorio */etc/nginx/conf.d/* del servidor MR un nuevo fichero de configuración, con un nombre identificativo, con la siguiente estructura:

```
server {
        listen 80;
        server_name NOMBRE(S)_DOMINIO_SERVICIO_WEB;

        location / {
                access_log off;
                proxy_pass http://DIRECCIÓN_IP_SERVIDOR_WEB_VIRTUAL;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
         }
}
~
"/etc/nginx/conf.d/NOMBRE_FICHERO.conf"
```

Donde ***NOMBRE(S)_DOMINIO_SERVICIO_WEB*** se refiere al nombre o nombres de dominio correspondientes al servicio web en cuestión, mientras que ***DIRECCIÓN_IP_SERVIDOR_WEB_VIRTUAL*** se refiere a la dirección IP del servidor virtual de alojamiento de dicho servicio web.

Una vez creado el fichero, se recargó el sistema de NGINX del servidor MR:

```
# nginx -s reload
```

Configuración del proxy inverso del servidor MR: Especificación de un servidor por defecto

Además, se decidió que se devolvería el código de estado 404 por defecto en caso de que se accediera a cualquier nombre de dominio asociado al servidor MR que no estuviera especificado en el *proxy* inverso.

Para ello se creó un nuevo fichero de configuración del *proxy* inverso llamado *wronghosts.conf* con la estructura siguiente:

```
server {
        listen 80 default_server;
        server_name _;
        return 404;
}
~
"/etc/nginx/conf.d/wronghosts.conf"
```

En el cual se puede observar la adición del parámetro *default_server* de la directiva *listen*.

Una vez modificado el fichero, se recargó el sistema de NGINX del servidor MR:

```
# nginx -s reload
```

ANEXO XVIII: CREACIÓN DE LA CARPETA COMPARTIDA "COPIASMV" EN EL SERVIDOR DE COPIAS DE SEGURIDAD B2

Para crear una carpeta compartida en el servidor B2 fue necesario acceder a su interfaz web y, mediante unas credenciales válidas, acceder al apartado "Configuración avanzada" (Ilustración 37).

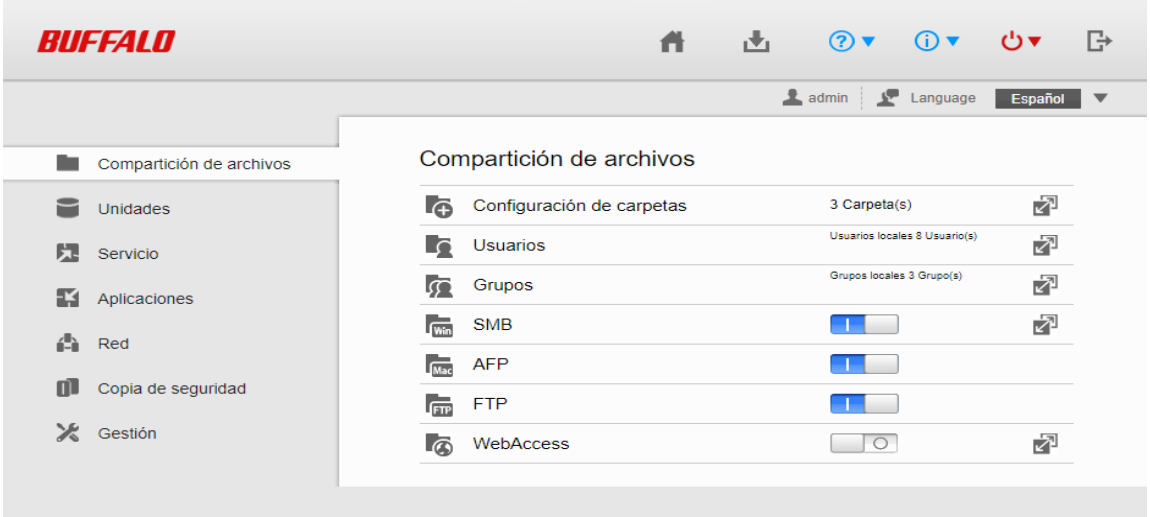


Ilustración 37: Página de inicio de la interfaz web del servidor de copias de seguridad B2

En esta página se clicó en el botón de "Configuración de carpetas" y se accedió a la lista de carpetas compartidas del servidor B2, desde donde se creó una carpeta compartida llamada *CopiasMV*, permitiéndosele las conexiones mediante el protocolo Samba para poder transferir a la misma los ficheros de las copias de seguridad, pero deshabilitando la opción de la papelera de reciclaje pues no se deseaba ocupar espacio innecesario (Ilustración 38).

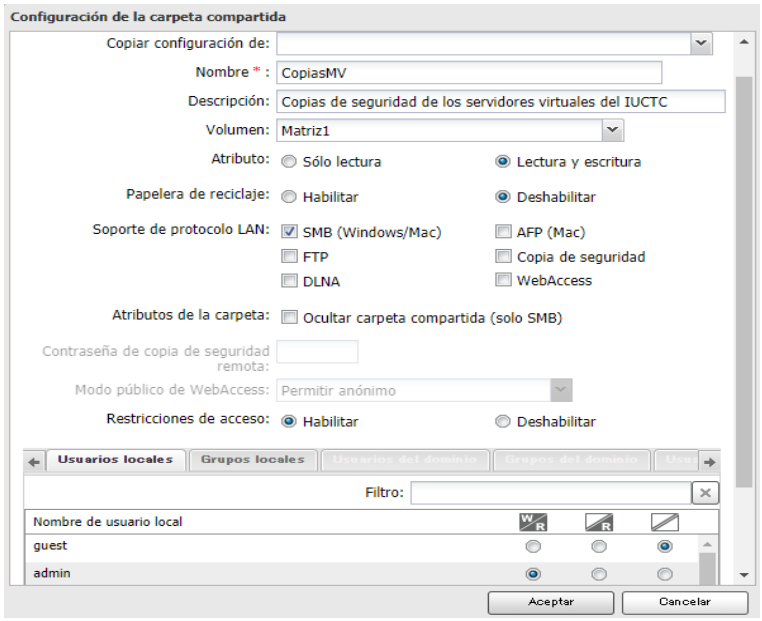


Ilustración 38: Página de creación de la carpeta compartida "CopiasMV" en el servidor de copias de seguridad B2

ANEXO XIX: DESARROLLO DE LOS *SCRIPTS* DEL SISTEMA DE COPIAS DE SEGURIDAD IMPLEMENTADO

Creación del directorio de almacenamiento de los ficheros del sistema de copias de seguridad: /usr/local/bin/vmbackuptools/

Antes que nada, fue necesario crear el directorio */usr/local/bin/vmbackuptools/* en el servidor MR para el almacenamiento de los ficheros del sistema de copias de seguridad:

```
# mkdir /usr/local/bin/vmbackuptools/
```

Creación del directorio de montaje de la carpeta compartida "CopiasMV": /mnt/vmbackups/

Además, fue necesario crear el directorio */mnt/vmbackups/* en el servidor MR como punto de montaje fijo de la carpeta compartida *CopiasMV* del servidor de copias de seguridad B2:

```
# mkdir /mnt/vmbackups/
```

Una vez creado este directorio, se procedió a crear y desarrollar cada uno de los *scripts* del sistema de copias de seguridad.

Creación del fichero de registro de los eventos del sistema de copias de seguridad: vmbackup.log

En primer lugar, se creó el fichero de registro *vmbackup.log* del sistema de copias de seguridad bajo el directorio */usr/local/bin/vmbackuptools/* del servidor MR:

```
# touch /usr/local/bin/vmbackuptools/vmbackup.log
```

Creación y desarrollo del script principal de copias de seguridad: vmbackup.sh

A continuación, se creó el *script* principal de copias de seguridad *vmbackup.sh* bajo el directorio */usr/local/bin/vmbackuptools/* del servidor MR:

```
# touch /usr/local/bin/vmbackuptools/vmbackup.sh
```

Y se le otorgó el permiso de ejecución:

```
# chmod +x /usr/local/bin/vmbackuptools/vmbackup.sh
```

Una vez desarrollado el *script*, este realizaría las siguientes acciones cada vez que fuera a ser ejecutado:

1) Establecer el fichero de registro *vmbackup.log* para registrar cada una de las acciones realizadas por el *script*.
2) Crear un directorio temporal en el servidor MR.
3) Apagar todos aquellos servidores web virtuales en ejecución.
4) Copiar las imágenes de disco de cada servidor web virtual en el directorio temporal. Este proceso fue pensado para que la indisponibilidad de cada servicio web fuera mínima, siendo la duración de alrededor de 2 o 3 minutos para la mayoría de las imágenes de disco y 12 minutos para las imágenes de disco de mayor tamaño.
5) Volver a iniciar los servidores web virtuales que estuvieran originalmente en ejecución.
6) Montar la carpeta compartida *CopiasMV* del servidor de copias de seguridad B2 en el directorio */mnt/vmbackups/* del servidor MR.
7) Crear un nuevo directorio en la carpeta compartida *CopiasMV* para las copias de seguridad a realizar.
8) Copiar los ficheros XML de cada servidor web virtual al directorio creado en la carpeta compartida *CopiasMV*.
9) Reducir las imágenes de disco anteriormente almacenadas en el directorio temporal creado en el servidor MR. Esto reduciría el tamaño de las imágenes de disco dando como resultado una optimización del espacio ocupado y el tiempo de transferencia.
10) Comprimir y mover cada imagen de disco del directorio temporal al directorio creado en la carpeta compartida *CopiasMV*. Para esto se utilizó la herramienta *pigz*, debido a su gran rendimiento y velocidad de transferencia con respecto a otras herramientas de compresión de ficheros.
11) Eliminar permanentemente el directorio temporal y todo su contenido del servidor MR.
12) Por último, desmontar la carpeta compartida *CopiasMV*.

El código fuente del *script vmbackup.sh* se expone a continuación:

```
!/bin/bash

# Se establece el fichero log del script
LOGFILE=/usr/local/bin/vmbackuptools/vmbackup.log

echo "$(date "+%d/%m/%Y %T") : Iniciado el programa de copias de seguridad de todos los
dominios." >> $LOGFILE
```

```
# Se crea un directorio temporal y se le otorga los permisos necesarios
echo "$(date "+%d/%m/%Y %T") : Creando un directorio temporal..." >> $LOGFILE
tmpdir=$(mktemp -d /tmp/vmbackup.XXXXXXXXXX)
chmod 755 $tmpdir
echo "$(date "+%d/%m/%Y %T") : El directorio temporal \"$tmpdir\" ha sido creado con
éxito." >> $LOGFILE

# Se cambia de directorio al directorio temporal
cd $tmpdir

# Se recorren todos los ficheros XML de las máquinas virtuales existentes
for filename_path in /etc/libvirt/qemu/*.xml; do
    # Se guarda el nombre de la máquina virtual
    vmname=$(basename "$filename_path" | cut -f 1 -d ".")

    #
    # COMENTAR PARA INCLUIR LA MÁQUINA MOODLE EN LA COPIA DE SEGURIDAD
    #
    if [ $vmname == "Moodle" ]; then
        continue;
    fi

    #
    # COMENTAR PARA INCLUIR LA MÁQUINA INTRANET EN LA COPIA DE SEGURIDAD
    #
    if [ $vmname == "Intranet" ]; then
        continue;
    fi

    # Se guarda el estado original de cada máquina
    orig_shutdown=$(virsh list --state-shutoff | sed '1d' | awk -F ' ' '{print $2}' | grep
-w $vmname)

    # Se apagan las máquinas en ejecución
    if [ ! $orig_shutdown ]; then
        virsh shutdown --domain $vmname >> $LOGFILE 2>&1
        sleep 25
        confirm_shutdown=$(virsh list --state-shutoff | sed '1d' | awk -F ' ' '{print $2}'
| grep -w $vmname)
        if [ ! $confirm_shutdown ]; then
            virsh destroy --domain $vmname >> $LOGFILE 2>&1
            sleep 25
        fi
        echo "$(date "+%d/%m/%Y %T") : Dominio $vmname apagado." >> $LOGFILE
    fi

    msg1="$(date "+%d/%m/%Y %T") : Copiando imagen/es de disco del dominio "
    msg2=" al directorio temporal \"$tmpdir\"..."
    msg3="$msg1$vmname$msg2"
    echo $msg3 >> $LOGFILE

    # Se obtienen las imágenes de disco de cada máquina del directorio local
/var/lib/libvirt/images/ y se copian al directorio temporal
    virsh domblklist $vmname | grep images | while read -r line
    do
        disk_img_path=$(echo "$line" | awk '{print $2}')
        disk_img_name=$(basename "$disk_img_path")

        cp $disk_img_path $disk_img_name >> $LOGFILE 2>&1
    done

    msg1="$(date "+%d/%m/%Y %T") : Copia completada. Volviendo a iniciar el dominio "
    msg2="..."
    msg3="$msg1$vmname$msg2"
    echo $msg3 >> $LOGFILE

    # Se vuelven a iniciar las máquinas que estaban en ejecución originalmente
    if [ ! $orig_shutdown ]; then
        virsh start --domain $vmname >> $LOGFILE 2>&1
        sleep 25
    fi
done

# Se monta el servidor de copias de seguridad en el directorio /mnt/vmbackups/
mount -t cifs //DIRECCIÓN_IP_SERVIDOR_B2/CopiasMV/ -o
user=NOMBRE_USUARIO,pass=CONTRASEÑA,vers=1.0 /mnt/vmbackups/ >> $LOGFILE 2>&1
```

```
# Se cambia de directorio al directorio del servidor de copias de seguridad
cd /mnt/vmbackups/

# Se guarda y se crea una ruta utilizando la fecha del día actual
year=$(date +"%Y")
month=$(date +"%m")
day=$(date +"%d")

if [ ! -d $year ]; then
    mkdir $year
fi

cd $year

if [ ! -d $month ]; then
    mkdir $month
fi

cd $month

if [ ! -d $day ]; then
    mkdir $day
fi

# Se cambia de directorio al directorio del día actual
cd $day

# Se vuelve a recorrer todos los ficheros XML de las máquinas virtuales existentes
for filename_path in /etc/libvirt/qemu/*.xml; do
    # Se guarda el nombre de cada máquina
    vmname=$(basename "$filename_path" | cut -f 1 -d ".")

    #
    # COMENTAR PARA INCLUIR LA MÁQUINA MOODLE EN LA COPIA DE SEGURIDAD
    #
    if [ $vmname == "Moodle" ]; then
        continue;
    fi

    #
    # COMENTAR PARA INCLUIR LA MÁQUINA INTRANET EN LA COPIA DE SEGURIDAD
    #
    if [ $vmname == "Intranet" ]; then
        continue;
    fi

    # Se crea un directorio con el nombre de cada máquina
    mkdir $vmname

    # Se cambia de directorio al directorio de cada máquina
    cd $vmname

    msg1="$(date "+%d/%m/%Y %T") : Copiando archivo XML del dominio "
    msg2=" al servidor de copias de seguridad..."
    msg3="$msg1$vmname$msg2"
    echo $msg3 >> $LOGFILE

    # Se copia el fichero XML correspondiente a cada máquina
    cp -v $filename_path . >> $LOGFILE 2>&1

    msg1="$(date "+%d/%m/%Y %T") : Copiando imagen/es de disco del dominio "
    msg3="$msg1$vmname$msg2"
    echo $msg3 >> $LOGFILE


    # Se vuelve a obtener las imágenes de disco de cada máquina y se copian del directorio
temporal al servidor de copias de seguridad
    virsh domblklist $vmname | grep images | while read -r line; do
        # Se guarda el nombre y la extensión de cada imagen de disco
        disk_img_path=$(echo "$line" | awk '{print $2}')
        disk_img_name=$(basename "$disk_img_path")

        msg1="$(date "+%d/%m/%Y %T") : >> Reduciendo la imagen del disco "
        msg2=" del dominio "
        msg3="..."
        msg4="$msg1$disk_img_name$msg2$vmname"
        echo $msg4 >> $LOGFILE
```

```
        # Se reduce cada imagen de disco mediante la herramienta "virt-sparsify"
        virt-sparsify --in-place $tmpdir/$disk_img_name > /dev/null 2>&1

        msg1="$(date "+%d/%m/%Y %T") : >> Comprimiendo la imagen del disco "
        msg2=" del dominio "
        msg3="..."
        msg4="$msg1$disk_img_name$msg2$vmname"
        echo $msg4 >> $LOGFILE

        # Se comprime cada imagen de disco mediante la herramienta de compresión "pigz" y
se copian al servidor de copias de seguridad
        pigz -c $tmpdir/$disk_img_name > $disk_img_name.gz
    done

    msg1="$(date "+%d/%m/%Y %T") : Copia de seguridad de la/s imagen/es de disco del
dominio "
    msg2=" completada."
    msg3="$msg1$vmname$msg2"
    echo $msg3 >> $LOGFILE

    # Se cambia de directorio al directorio padre
    cd ..
done

# Se elimina por completo el directorio temporal y todo su contenido
echo "$(date "+%d/%m/%Y %T") : Eliminando el directorio temporal \"$tmpdir\"..." >>
$LOGFILE
rm -rfv $tmpdir >> $LOGFILE 2>&1
echo "$(date "+%d/%m/%Y %T") : El directorio temporal \"$tmpdir\" ha sido eliminado con
éxito." >> $LOGFILE

# Se cambia de directorio al directorio de montaje /mnt/
cd /mnt/

echo "$(date "+%d/%m/%Y %T") : Programa de copias de seguridad finalizado." >> $LOGFILE

# Se desmonta el servidor de copias de seguridad del directorio /mnt/vmbackups/
umount /mnt/vmbackups/ >> $LOGFILE 2>&1

exit 0
```

Creación y desarrollo del script de restauración de copias de seguridad: vmrestore.sh

Seguidamente, se creó el *script* interactivo de restauración de copias de seguridad *vmrestore.sh* bajo el directorio */usr/local/bin/vmbackuptools/* del servidor MR:

```
# touch /usr/local/bin/vmbackuptools/vmrestore.sh
```

Y se le otorgó el permiso de ejecución:

```
# chmod +x /usr/local/bin/vmbackuptools/vmrestore.sh
```

Una vez desarrollado el *script*, este realizaría las siguientes acciones cada vez que fuera a ser ejecutado:

1) Montar la carpeta compartida *CopiasMV* del servidor de copias de seguridad B2 en el directorio */mnt/vmbackups/* del servidor MR.
2) Requerir al usuario la ruta del fichero XML de la copia de seguridad del servidor web virtual a restaurar.
3) Si se procede a la restauración, apagar el servidor web virtual correspondiente a la copia de seguridad a restaurar.
4) Restaurar el fichero XML de la copia de seguridad elegida en el servidor MR.
5) Restaurar las imágenes de disco de la copia de seguridad elegida en el servidor MR.
6) Reducir las imágenes de disco del servidor web virtual restaurado.
7) Desmontar la carpeta compartida *CopiasMV*.
8) Por último, volver a iniciar el servidor web virtual restaurado en caso de que estuviera originalmente en ejecución.

El código fuente del *script vmrestore.sh* se expone a continuación:

```
#!/bin/bash

# Función que desmonta el servidor de copias de seguridad del directorio /mnt/vmbackups/ en caso de que se active la trampa
function umountserv {
    # Se cambia de directorio al directorio de montaje /mnt/
    cd /mnt

    # Se desmonta el servidor de copias de seguridad del directorio /mnt/vmbackups/
    umount /mnt/vmbackups/

    printf "\nServidor de copias de seguridad desmontado."

    printf "\n### ADIÓS ###\n"

    exit 0
}

# Trampa que atrapa las señales SIGINT, SIGTERM y SIGTSTP
trap umountserv 2 15 20
```

```
echo "### BIENVENIDO AL SCRIPT DE RESTAURACIÓN DE DOMINIOS DE KVM (PARA SALIR PULSE CTRL-C
Ó CTRL-Z) ###"

# Se monta el servidor de copias de seguridad en el directorio /mnt/vmbackups/
mount -t cifs //DIRECCIÓN_IP_SERVIDOR_B2/CopiasMV/ -o
user=NOMBRE_USUARIO,pass=CONTRASEÑA,vers=1.0 /mnt/vmbackups/

echo "Servidor de copias de seguridad montado."

# Se cambia de directorio al directorio del servidor de copias de seguridad
cd /mnt/vmbackups/

loop1=1

# Se muestra una interfaz simple para que el usuario pueda navegar hasta el directorio del
fichero XML de la máquina a restaurar desde el servidor de copias de seguridad
while [[ $loop1 -eq 1 ]]; do
    loop1=0

    loop2=1

    while [[ $loop2 -eq 1 ]]; do
        loop2=0

        read -e -p "Introduzca la ruta absoluta del fichero XML de la copia del dominio a
restaurar: " -i "/mnt/vmbackups/" filepath

        if [ ! -e $filepath ]; then
            loop2=1
            echo "El fichero no existe."
            continue
        fi

        if [ ! -f $filepath ]; then
            loop2=1
            echo "El fichero especificado no es un fichero regular."
            continue
        fi

        fullname=$(basename "$filepath")
        vmname="${fullname%.*}"
        extension="${fullname##*.}"

        if [ $extension != "xml" ]; then
            loop2=1
            echo "El fichero no es de tipo XML."
            continue
        fi
    done

    loop3=1


    # Se muestra una interfaz simple de confirmación de la operación
    while [[ $loop3 -eq 1 ]]; do
        loop3=0

        read -p "¿Está seguro de que quiere restaurar el dominio $vmname a partir del
fichero $filepath? [S/N] " confirm

        if [ $confirm == "N" ]; then
            echo "Restauración cancelada."

            loop1=1

            break
        fi

        if [ $confirm != "S" ]; then
            loop3=1
        fi
    done
done
```

```
# En caso de que se proceda a la restauración se guarda primero el estado original de la
máquina
orig_shutdown=$(virsh list --state-shutoff | sed '1d' | awk -F ' ' '{print $2}' | grep -w
$vmname)

# Se apaga la máquina en caso de encontrarse en ejecución
if [ ! $orig_shutdown ]; then
    virsh shutdown --domain $vmname

    sleep 25

    echo "Dominio $vmname apagado."
fi

# En caso de encontrarse un fichero XML con el mismo nombre en el directorio
/etc/libvirt/qemu/ se le añade la extensión ".old" a este último
if [ -e /etc/libvirt/qemu/$fullname ]; then
    mv /etc/libvirt/qemu/$fullname /etc/libvirt/qemu/$fullname.old
fi

# Se restaura el fichero XML de la máquina en cuestión y se le otorga los permisos
necesarios
cp -v $filepath /etc/libvirt/qemu/
chmod 0600 /etc/libvirt/qemu/$fullname

echo "El fichero XML del dominio $vmname ha sido restaurado con éxito en
/etc/libvirt/qemu/..."

echo "Descomprimiendo y copiando imagen/es de disco del dominio $vmname del servidor de
copias de seguridad a /var/lib/libvirt/images/..."

# Se guarda la ruta al directorio de la máquina
domaindir=$(dirname "$filepath")

# Se recorren todas las imágenes de disco (comprimidas) de la máquina a restaurar del
servidor de copias de seguridad
for diskpath in $domaindir/*.qcow2.gz; do

    # Se guarda el nombre de la imagen de disco (comprimida) así como el mismo nombre sin
la extensión ".gz"
    disknamegz=$(basename "$diskpath")
    diskname="${disknamegz%.*}"

    # En caso de encontrarse un fichero de imagen de disco con el mismo nombre (sin la
extensión ".gz") en el directorio local /var/lib/libvirt/images/ se le añade la extensión
".old" a este último
    if [ -e /var/lib/libvirt/images/$diskname ]; then
        mv /var/lib/libvirt/images/$diskname /var/lib/libvirt/images/$diskname.old
    fi

    # Se descomprime cada imagen de disco mediante la herramienta de compresión "unpigz" y
se copian al directorio local /var/lib/libvirt/images/ y se le otorga los permisos
necesarios
    unpigz -cv $diskpath > /var/lib/libvirt/images/$diskname
    chmod 0600 /var/lib/libvirt/images/$diskname

    # Se reducen las imágenes de disco de la máquina restaurada (esto se debe a que durante
la descompresión se pierde la reducción)
    virt-sparsify --in-place /var/lib/libvirt/images/$diskname
done

echo "La/s imagen/es de disco del dominio $vmname ha/n sido restaurada/s con éxito."
echo "Restauración completada."

# Se cambia de directorio al directorio de montaje /mnt/
cd /mnt

# Se desmonta el servidor de copias de seguridad del directorio /mnt/vmbackups/
umount /mnt/vmbackups/

echo "Servidor de copias de seguridad desmontado."

# Se vuelve a iniciar la máquina en caso de que estuviera en ejecución originalmente
if [ ! $orig_shutdown ]; then
        virsh start --domain $vmname
fi
```

```
printf "\n### ADIÓS ###\n"

exit 0
```

Creación y desarrollo del script de limpieza de copias de seguridad: vmbackup_clean.sh

Por último, se creó el *script* de limpieza de copias de seguridad *vmbackup_clean.sh* bajo el directorio */usr/local/bin/vmbackuptools/* del servidor MR:

```
# touch /usr/local/bin/vmbackuptools/vmbackup_clean.sh
```

Y se le otorgó el permiso de ejecución:

```
# chmod +x /usr/local/bin/vmbackuptools/vmbackup_clean.sh
```

Una vez desarrollado el *script*, este realizaría las siguientes acciones cada vez que fuera a ser ejecutado:

1) Establecer el fichero de registro *vmbackup.log* para registrar cada una de las acciones realizadas por el *script*.
2) Montar la carpeta compartida *CopiasMV* del servidor de copias de seguridad B2 en el directorio */mnt/vmbackups/* del servidor MR.
3) Eliminar todas las copias de seguridad guardadas del día de hace cierto periodo de tiempo desde la fecha de ejecución del *script*.
4) Por último, desmontar la carpeta compartida *CopiasMV*.

El código fuente del *script vmbackup_clean.sh* se expone a continuación:

```
#!/bin/bash

# Se establece el fichero log del script
LOGFILE=/usr/local/bin/vmbackuptools/vmbackup.log

# Se monta el servidor de copias de seguridad en el directorio /mnt/vmbackups/
mount -t cifs //DIRECCIÓN_IP_SERVIDOR_B2/CopiasMV/ -o
user=NOMBRE_USUARIO,pass=CONTRASEÑA,vers=1.0 /mnt/vmbackups/ >> $LOGFILE 2>&1

echo "$(date "+%d/%m/%Y %T") : Borrando copias de seguridad excedentes..." >> $LOGFILE
# Se eliminan los ficheros de un mes de antigüedad. CAMBIAR EN CASO DE QUE SE CAMBIE LA
PERIODICIDAD DE LAS COPIAS DE SEGURIDAD.
find /mnt/vmbackups -type f -mtime +30 -exec rm -f {} \;

# Se eliminan los directorios vacíos.
find /mnt/vmbackups -type d -empty -delete

# Se desmonta el servidor de copias de seguridad del directorio /mnt/vmbackups/
umount /mnt/vmbackups/ >> $LOGFILE 2>&1

exit 0
```

ANEXO XX: PROGRAMACIÓN DE LA EJECUCIÓN DE LOS *SCRIPTS* DEL SISTEMA DE COPIAS DE SEGURIDAD IMPLEMENTADO

Programación de la ejecución del script vmbackup.sh mediante Cron

Se acordó que la ejecución del programa o script principal del sistema de copias de seguridad se realizaría, temporalmente, a las 03:00 de cada día, ya que se observó que era una hora en la que la cantidad de accesos a los servicios web era muy baja.

Se calculó y comprobó que la duración del proceso completo oscilaba entre la hora y media y las dos horas, por lo que se pudo determinar que el proceso era perfectamente viable.

Para programar la ejecución del *script* a dicha hora se utilizó, una vez más, el administrador de procesos de Cron. Para ello se modificó el fichero *crontab* del servidor MR:

```
# crontab -e
```

Añadiendo, en este caso, la siguiente línea:

```
0 3 * * * /usr/local/bin/vmbackuptools/vmbackup.sh
```

Programación de la ejecución del script vmbackup_clean.sh mediante Cron

Además, fue necesario programar la ejecución del *script* de limpieza de las copias de seguridad del sistema de copias de seguridad, ya que tarde o temprano el disco del servidor de copias de seguridad B2 acabaría quedándose sin espacio de almacenamiento.

Por ello, se programó la ejecución de dicho *script* justo cinco minutos antes de la ejecución del *script* principal de copias de seguridad (*vmbackup.sh*), es decir, a las 02:55 de cada día, garantizándose de esta forma la existencia de espacio libre en el disco del servidor B2 para el posterior almacenamiento de las copias de seguridad que realizaría el *script* principal.

Para programar la ejecución del *script* a dicha hora se utilizó también el administrador de procesos de Cron. Para ello se modificó el fichero *crontab* del servidor MR:

```
# crontab -e
```

Añadiendo, en este caso, la siguiente línea:

```
# 55 2 * * * /usr/local/bin/vmbackuptools/vmbackup_clean.sh
```

ANEXO XXI: CONFIGURACIÓN DEL SERVIDOR MD

Modificación del nombre de host

En primer lugar, se modificó el nombre de *host* del servidor MD:

```
# hostnamectl set-hostname NOMBRE_HOST.ciber.ulpgc.es
```

Modificación de la configuración de red

A continuación, se configuró la interfaz de red del servidor MD, de forma que tuviera asignada una dirección IP estática conocida para su fácil identificación en la red interna del centro. Para ello se especificaron en el fichero */etc/sysconfig/network-scripts/ifcfg-enp5s0f0* del servidor los siguientes parámetros:

```
…
BOOTPROTO=none
…
ONBOOT=yes
ZONE=public
IPADDR=DIRECCIÓN_IP_PRIVADA_SERVIDOR_MD
PREFIX=PREFIJO_MÁSCARA_SUBRED
GATEWAY=DIRECCIÓN_IP_PUERTA_DE_ENLACE_RED_PRIVADA
DNS1=193.145.138.100
DNS2=193.145.138.200
~
"/etc/sysconfig/network-scripts/ifcfg-enp5s0f0"
```

Una vez guardados los cambios, se reinició el servidor MD:

```
# reboot
```

Actualización del sistema

Debido a la reciente reinstalación del sistema operativo en el servidor MD fue necesario actualizar todos los paquetes del nuevo sistema:

```
# yum update -y
```

Instalación de wget

También se requirió la instalación de la herramienta *wget*:

```
# yum install wget -y
```

Habilitación de servicio mediante FirewallD: Servicio HTTP

Para permitir las conexiones entrantes al servidor web primero fue necesario habilitar, mediante una regla, el protocolo HTTP en el cortafuegos del sistema a través de *FirewallD*:

```
# firewall-cmd --zone=public --add-service=http --permanent
# firewall-cmd --reload
```

Instalación de Apache

Se instaló el servidor web de Apache:

```
# yum install httpd -y
```

Se inició su servicio:

```
# systemctl start httpd
```

Y, además, se configuró para que se iniciara automáticamente con el arranque del servidor:

```
 # systemctl enable httpd
```

Instalación de MariaDB

Se instaló el sistema de bases de datos de MariaDB:

```
# yum install mariadb-server -y
```

Se inició su servicio:

```
# systemctl start mariadb
```

Se configuró para que se iniciara automáticamente con el arranque del servidor:

```
 # systemctl enable mariadb
```

Y, además, se procedió a realizar una instalación segura de dicho sistema:

```
# /usr/bin/mysql_secure_installation
```

Instalación de PHP

Seguidamente, se procedió a la instalación de PHP:

```
# yum install php -y
```

Instalación de Zabbix y su agente

Una vez instaladas las herramientas necesarias, se procedió a la instalación del sistema de monitorización de Zabbix. Para ello, en primer lugar, fue necesario instalar el repositorio de la última versión de Zabbix:

```
# rpm -Uvh RUTA_WEB_REPOSITORIO_ZABBIX
```

A continuación, se limpiaron todas las entradas de los repositorios activos en el momento de la memoria caché:

```
# yum clean all
```

Seguidamente, se instaló el sistema de monitorización de Zabbix y su interfaz web con soporte para bases de datos MySQL:

```
# yum install zabbix-server-mysql zabbix-web-mysql -y
```

Además, se instaló el agente de Zabbix, que permitiría al servidor recolectar datos sobre su propio estado de servicio:

```
# yum install zabbix-agent -y
```

Creación de la base de datos de Zabbix

Una vez instalado el sistema de monitorización, fue necesario crear una nueva base de datos para Zabbix.

Para ello primero se accedió a la interfaz de MariaDB:

```
# mysql -u NOMBRE_USUARIO_MYSQL -p
```

Una vez dentro, se creó la base de datos de la siguiente manera:

```
MariaDB [(none)]> CREATE DATABASE zabbix DEFAULT CHARACTER SET utf8 COLLATE utf8_bin;
```

Seguidamente se creó un usuario específico para dicha base de datos:

```
MariaDB [(none)]> CREATE USER 'NOMBRE_USUARIO_BD_ZABBIX'@'localhost' IDENTIFIED BY 'NUEVA_CONTRASEÑA_USUARIO_BD_ZABBIX';
```

Se le otorgaron al usuario todos los privilegios sobre la base de datos:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON zabbix.* TO 'NOMBRE_USUARIO_BD_ZABBIX'@'localhost' IDENTIFIED BY 'CONTRASEÑA_USUARIO_BD_ZABBIX' WITH GRANT OPTION;
```

Se recargó la tabla de privilegios del sistema de bases de datos:

```
MariaDB [(none)]> FLUSH PRIVILEGES;
```

Y se cerró la sesión de la terminal de MariaDB:

```
MariaDB [(none)]> EXIT;
```

Importación del esquema y los datos iniciales de Zabbix en la base de datos

Para importar el esquema y datos iniciales de Zabbix, provistos en un fichero descargado durante la instalación, en la base de datos de Zabbix fue necesario utilizar la herramienta *zcat* debido a que dicho fichero estaba comprimido:

```
# zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -u NOMBRE_USUARIO_BD_ZABBIX -p zabbix
```

Para que el servidor de monitorización usara esta base de datos se requirió especificar la contraseña de dicha base de datos en el fichero de configuración */etc/zabbix/zabbix_server.conf* del servidor:

```
…
# Mandatory: no
# Default
DBPassword=CONTRASEÑA_USUARIO_BD_ZABBIX
…
~
"/etc/zabbix/zabbix_server.conf"
```

Configuración de PHP para Zabbix

La interfaz web de Zabbix está escrita en PHP por lo que requiere algunas configuraciones especiales.

En este caso, para asegurar que Zabbix utilizara la hora adecuada, fue necesario establecer la zona horaria apropiada. Por ello se modificó el fichero de configuración */etc/httpd/conf.d/zabbix.conf* para añadir la siguiente línea:

```
…
<IfModule mod_php5.c>
        …
        php_value date.timezone Atlantic/Canary
</IfModule>
~
"/etc/httpd/conf.d/zabbix.conf"
```

Reinicio de Apache

Una vez realizados y guardados los cambios, se reinició el servidor web de Apache:

```
# systemctl restart httpd
```

Habilitación de puerto: Puerto 10051/tcp

Debido a que los agentes de Zabbix se comunican y envían datos al servidor (en este caso, al servidor MD) por el puerto 10051 fue necesario habilitarlo, mediante una regla, en el cortafuegos del sistema a través de *FirewallD*:

```
# firewall-cmd --zone=public --add-port=10051/tcp --permanent
# firewall-cmd --reload
```

Inicio de Zabbix y su agente

A continuación, se inició el sistema de monitorización de Zabbix:

```
# systemctl start zabbix-server
```

Y su agente:

```
# systemctl start zabbix-agent
```

Automatización del inicio de Zabbix y su agente

Seguidamente, se automatizó el inicio del sistema de monitorización de Zabbix:

```
# systemctl enable zabbix-server
```

Así como el de su agente:

```
# systemctl enable zabbix-agent
```

Instalación de la interfaz web de gestión de Zabbix

A continuación, se instaló la interfaz web de Zabbix que permitiría gestionar el sistema de monitorización. Para ello, se inició el navegador del propio servidor MD y se accedió a la dirección *http://**DIRECCIÓN_IP_SERVIDOR_MD**/zabbix/*, mostrando un mensaje de bienvenida.

En la siguiente pantalla, se mostró una tabla de todos los prerrequisitos necesarios para la instalación de la interfaz web (Ilustración 39).

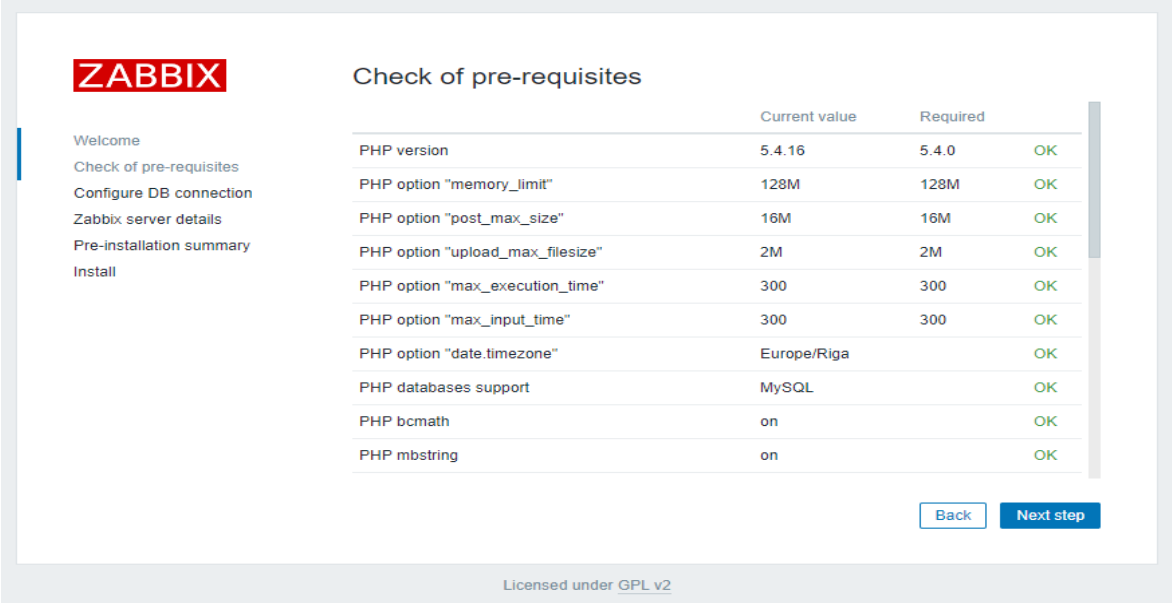


Ilustración 39: Ventana de prerrequisitos para la instalación de Zabbix

Se aseguró que todos los valores de dicha tabla estuvieran realmente en orden. Para ello se modificó el fichero */etc/php.ini* del servidor MD y se cambió el valor de algunos parámetros:

```
…
post_max_size = 16M
…
upload_max_filesize = 2M
…
max_execution_time = 300
…
max_input_time = 300
…
memory_limit = 128M
…
date.timezone = Atlantic/Canary
…
~
“/etc/php.ini”
```

Una vez que se verificó que se cumplían todos los prerrequisitos se accedió a la siguiente pantalla y se completó el formulario mostrado con la información de conexión a la base de datos de Zabbix (Ilustración 40).

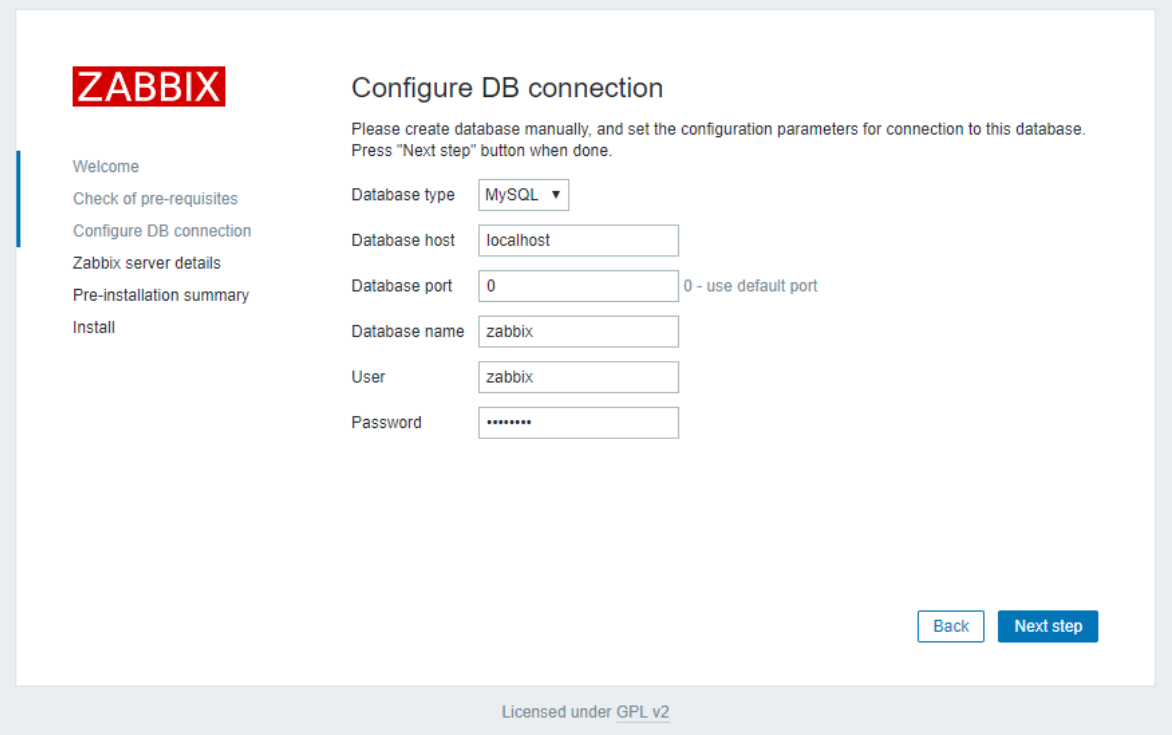


Ilustración 40: Ventana de configuración de la conexión del servidor Zabbix con su base de datos

En la pantalla posterior se mantuvieron los detalles del servidor de Zabbix mostrados por defecto (Ilustración 41).

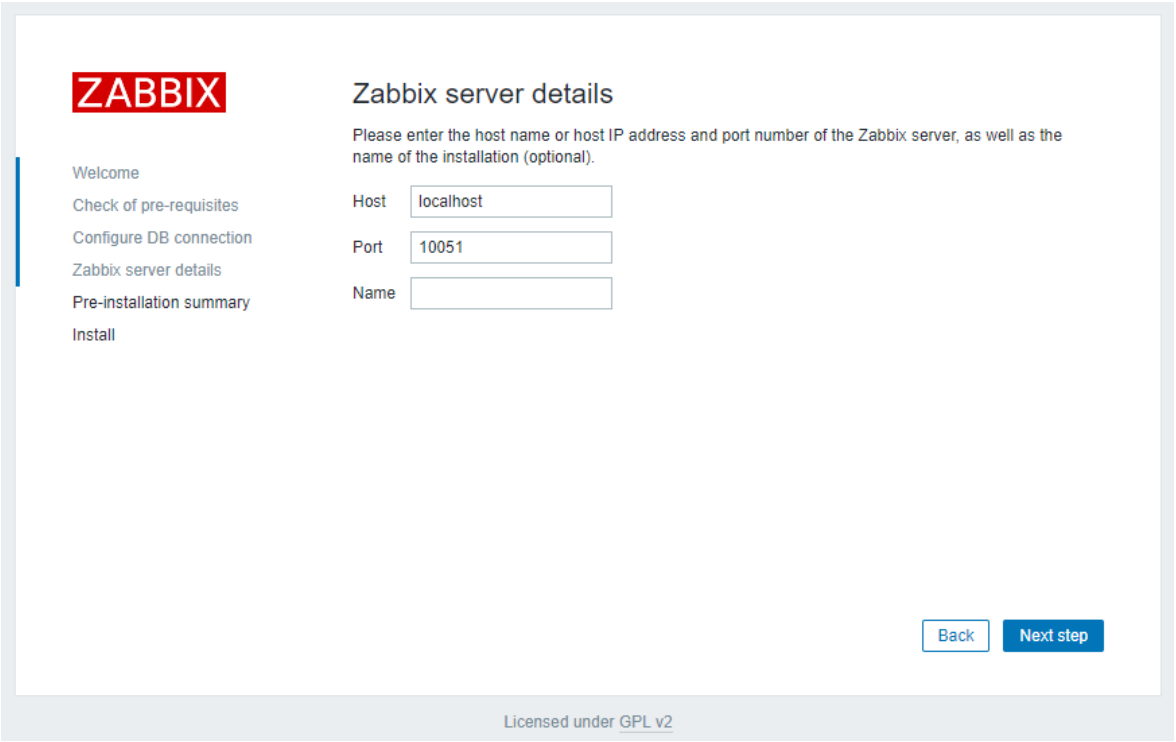


Ilustración 41: Ventana de configuración de detalles del servidor Zabbix

En la siguiente pantalla se mostró un resumen de los parámetros de configuración de la instalación (Ilustración 42).

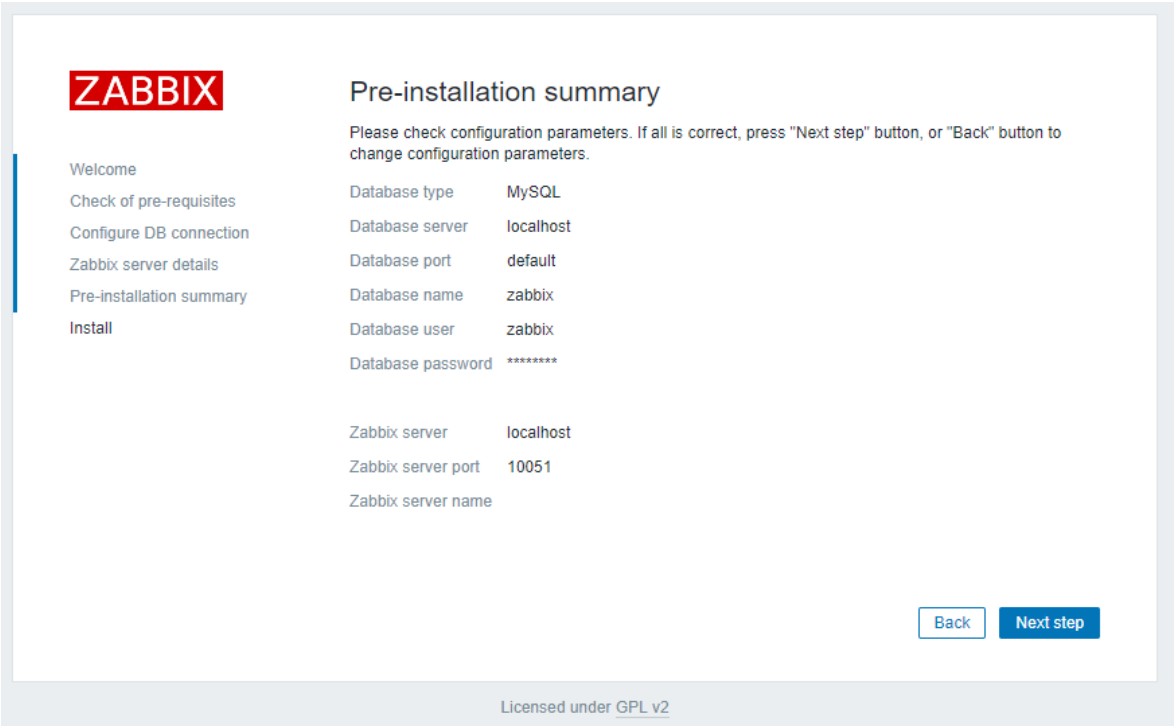


Ilustración 42: Ventana de resumen de los parámetros de configuración de Zabbix

Una vez que se confirmó que la configuración se había realizado correctamente se finalizó la instalación de la interfaz web de Zabbix.

Instalación y configuración del agente de Zabbix en el servidor MR

Posteriormente, para poder monitorizar el servidor MR, fue necesario instalar y configurar en este el agente de software de Zabbix, que recolectaría información del estado del servicio de dicho servidor y se lo enviaría al servidor MD, en el cual estaba instalado el sistema de monitorización.

Para ello, primero se accedió como administrador al servidor MR y se instaló el repositorio de la última versión de Zabbix:

```
# rpm -Uvh RUTA_WEB_REPOSITORIO_ZABBIX
```

A continuación, se limpiaron todas las entradas de los repositorios activos en el momento de la memoria caché:

```
# yum clean all
```

Seguidamente, se instaló el agente de Zabbix:

```
# yum install zabbix-agent -y
```

Tras la instalación del agente, se decidió asegurar la conexión entre el agente y el servidor utilizando una clave previamente compartida (*Pre-shared Key*, PSK) [45]. Para ello, primero se generó una clave PSK en el servidor MR:

```
# sh -c “openssl rand -hex 32 > /etc/zabbix/zabbix_agentd.psk”
```

Una vez generada, se modificó el fichero de configuración del agente de Zabbix */etc/zabbix/zabbix_agentd.conf* del servidor MR y se especificaron los siguientes parámetros:

```
…
### Option: Server
#       List of comma delimited IP addresses (or hostnames) of Zabbix servers.
#       Incoming connections will be accepted only from the hosts listed here.
#       If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally.
#
# Mandatory: no
# Default:
# Server=

Server=DIRECCIÓN_IP_SERVIDOR_MD
…
### Option: TLSConnect
#       How the agent should connect to server or proxy. Used for active checks.
#       Only one value can be specified:
#               unencrypted - connect without encryption
#               psk         - connect using TLS and a pre-shared key
#               cert        - connect using TLS and a certificate
#
# Mandatory: yes, if TLS certificate or PSK parameters are defined (even for 'unencrypted' connection)
# Default:
# TLSConnect=unencrypted

TLSConnect=psk
…
### Option: TLSAccept
#       What incoming connections to accept.
#       Multiple values can be specified, separated by comma:
#               unencrypted - accept connections without encryption
#               psk         - accept connections secured with TLS and a pre-shared key
#               cert        - accept connections secured with TLS and a certificate
#
# Mandatory: yes, if TLS certificate or PSK parameters are defined (even for 'unencrypted' connection)
# Default:
# TLSAccept=unencrypted

TLSAccept=psk
…
### Option: TLSPSKIdentity
#       Unique, case sensitive string used to identify the pre-shared key.
#
# Mandatory: no
# Default:
# TLSPSKIdentity=

TLSPSKIdentity=PSK 001
…
### Option: TLSPSKFile
#       Full pathname of a file containing the pre-shared key.
#
# Mandatory: no
# Default:
# TLSPSKFile=

TLSPSKFile=/etc/zabbix/zabbix_agentd.psk
…
~
"/etc/zabbix/zabbix_agentd.conf"
```

Habilitación de puerto en el servidor MR mediante Iptables: Puerto 10050/tcp

Debido a que el sistema de monitorización de Zabbix (servidor MD) se comunica con sus agentes (servidor MR) por el puerto 10050 fue necesario habilitarlo en el cortafuegos del servidor MR a través de *Iptables* de forma que el fichero de configuración */etc/sysconfig/iptables* tuviera, finalmente, el siguiente contenido:

```
*filter

:INPUT DROP [0:0]

:FORWARD DROP [0:0]

:OUTPUT ACCEPT [30:3059]

-A INPUT -i lo -m state --state NEW -j ACCEPT

-A INPUT -p icmp -m icmp --icmp-type 8 -m state --state NEW -j ACCEPT

-A INPUT -p udp -m udp -m multiport --dports 80,443 -m state --state NEW -j ACCEPT

-A INPUT -p tcp -m tcp -m multiport --dports 22,80,443 -m state --state NEW -j ACCEPT

-A INPUT -i em2 -p tcp -m tcp --dport 3389 -m state --state NEW -j ACCEPT

-A INPUT -i virbr0 -p udp -m udp -m multiport --dports 53,67 -j ACCEPT

-A INPUT -i virbr0 -p tcp -m tcp -m multiport --dports 53,67 -j ACCEPT

-A INPUT -s DIRECCIÓN_IP_RED_INTERNA/MÁSCARA_SUBRED -p tcp -m tcp --dport 10050 -m state --state NEW -j
ACCEPT

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

-A FORWARD -i virbr0 -o virbr0 -j ACCEPT

-A FORWARD -s DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED -i virbr0 -j ACCEPT

-A FORWARD -d DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED -o virbr0 -m state --state RELATED,ESTABLISHED -
j ACCEPT

COMMIT

*nat

:PREROUTING ACCEPT [21:1500]

:INPUT ACCEPT [18:1086]

:OUTPUT ACCEPT [2:120]

:POSTROUTING ACCEPT [2:120]

-A POSTROUTING -s DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED ! -d
DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED -p udp -m udp -j MASQUERADE --to-ports 1024-65535

-A POSTROUTING -s DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED ! -d
DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED -p tcp -m tcp -j MASQUERADE --to-ports 1024-65535

-A POSTROUTING -s DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED ! -d
DIRECCIÓN_IP_RED_VIRTUAL/MÁSCARA_SUBRED -j MASQUERADE

COMMIT

…

~

"/etc/sysconfig/iptables"
```

Inicio del agente de monitorización de Zabbix en el servidor MR

A continuación, se inició el agente de monitorización de Zabbix:

```
# systemctl start zabbix-agent
```

Automatización del inicio del agente de monitorización de Zabbix en el servidor MR

Seguidamente, se automatizó el inicio del agente de monitorización de Zabbix:

```
# systemctl enable zabbix-agent
```

Configuración para la monitorización del servidor MR mediante Zabbix

Una vez fue instalado y configurado el agente de monitorización de Zabbix en el servidor MR fue necesario registrarlo en el sistema de monitorización instalado en el servidor MD.

Para ello se ingresó a la interfaz web del sistema de Zabbix y se accedió a *Configuración* > *Equipos* (Ilustración 43).

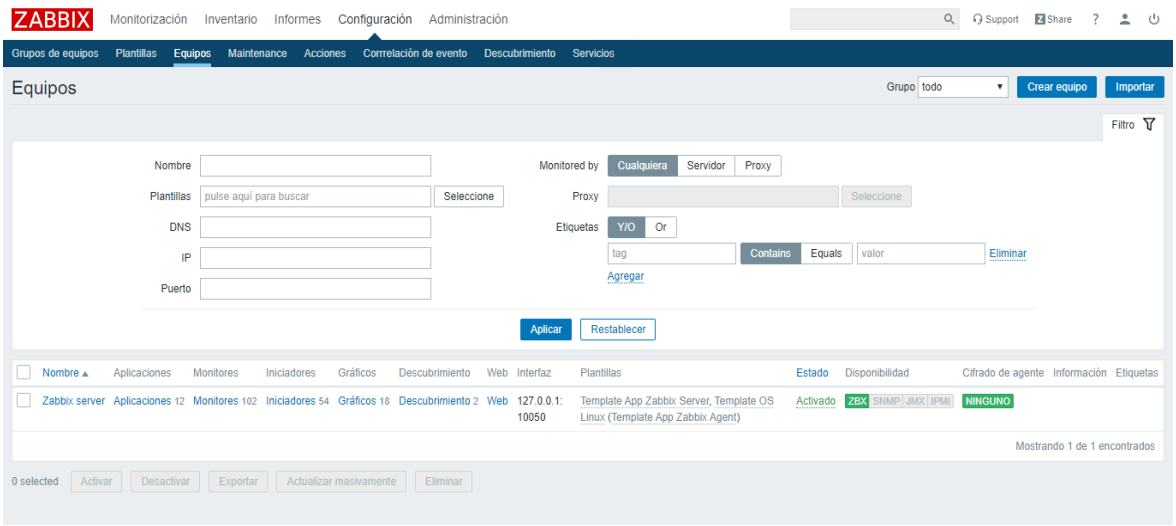


Ilustración 43: Interfaz web de Zabbix - Lista de equipos monitorizados

Luego se pulsó el botón "Crear Equipo" situado en la esquina superior derecha de la pantalla, mostrándose así la página de creación de un nuevo equipo (Ilustración 44).

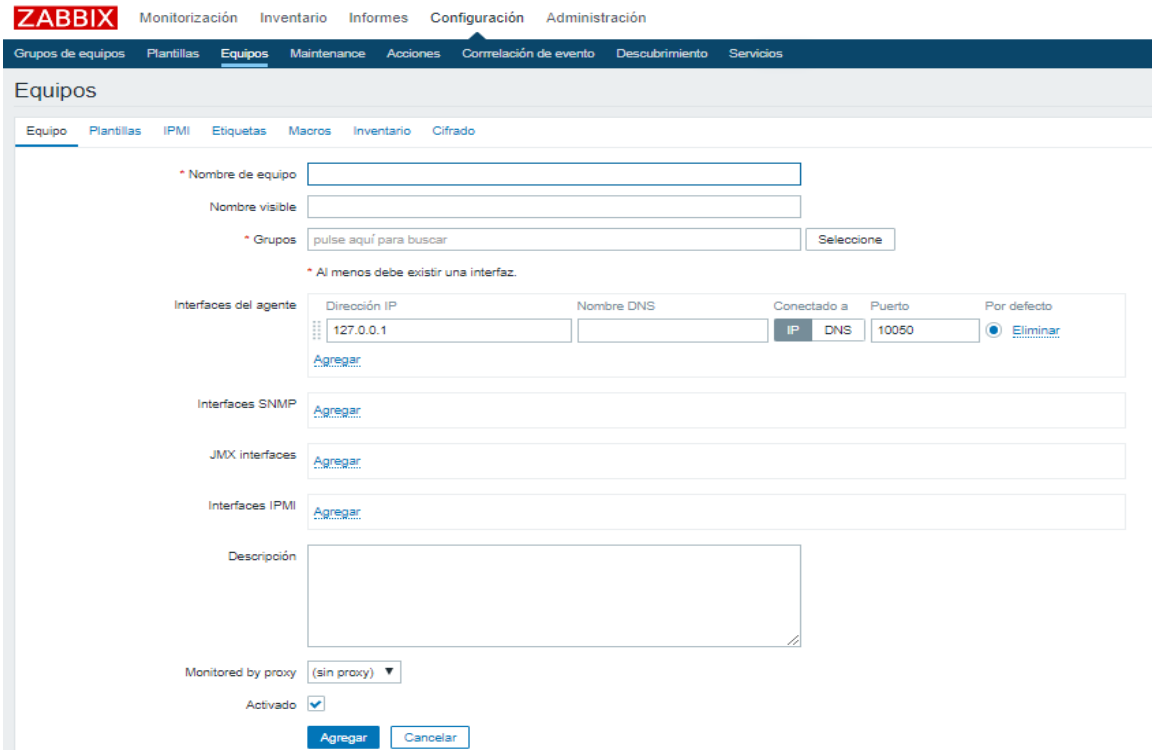


Ilustración 44: Interfaz web de Zabbix - Página de creación de un nuevo equipo I

En dicha página se introdujo el nombre de equipo y la dirección IP del servidor MR, en el cual estaba instalado el agente, y se añadió el equipo a los grupos "Linux Servers" e "Hypervisors".

Una vez hecho esto, se clicó en la pestaña "Plantillas" del propio formulario (Ilustración 45).

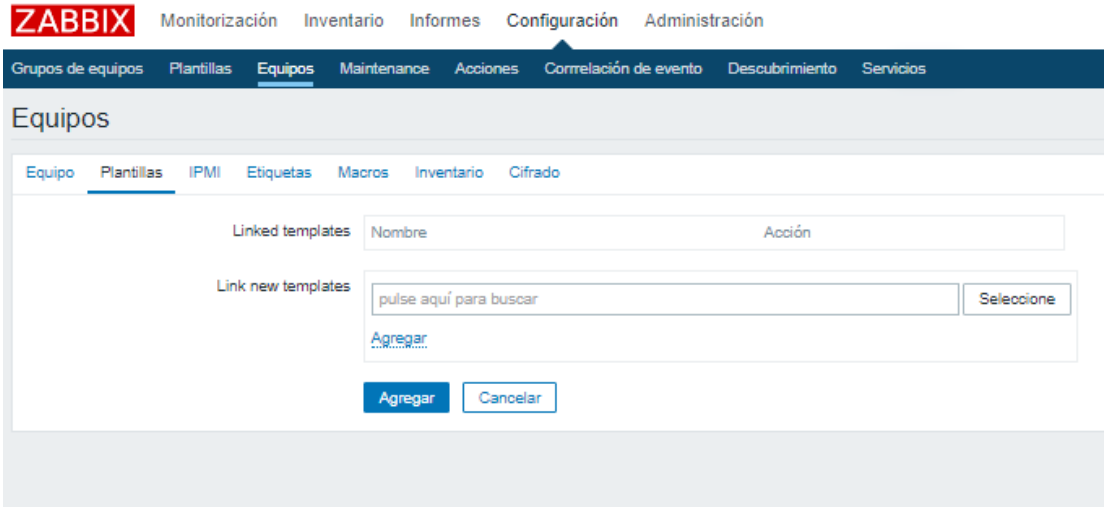


Ilustración 45: Interfaz web de Zabbix - Página de creación de un nuevo equipo II

Para facilitar la monitorización del servidor MR, se especificó una plantilla (*template*) con un conjunto de recursos de monitorización ya preestablecidos, en este caso, se decidió utilizar la plantilla denominada "Template OS Linux" pues era la que mejor se ajustaba al sistema instalado en el servidor.

A continuación, se accedió a la pestaña "Cifrado" del mismo formulario (Ilustración 46).

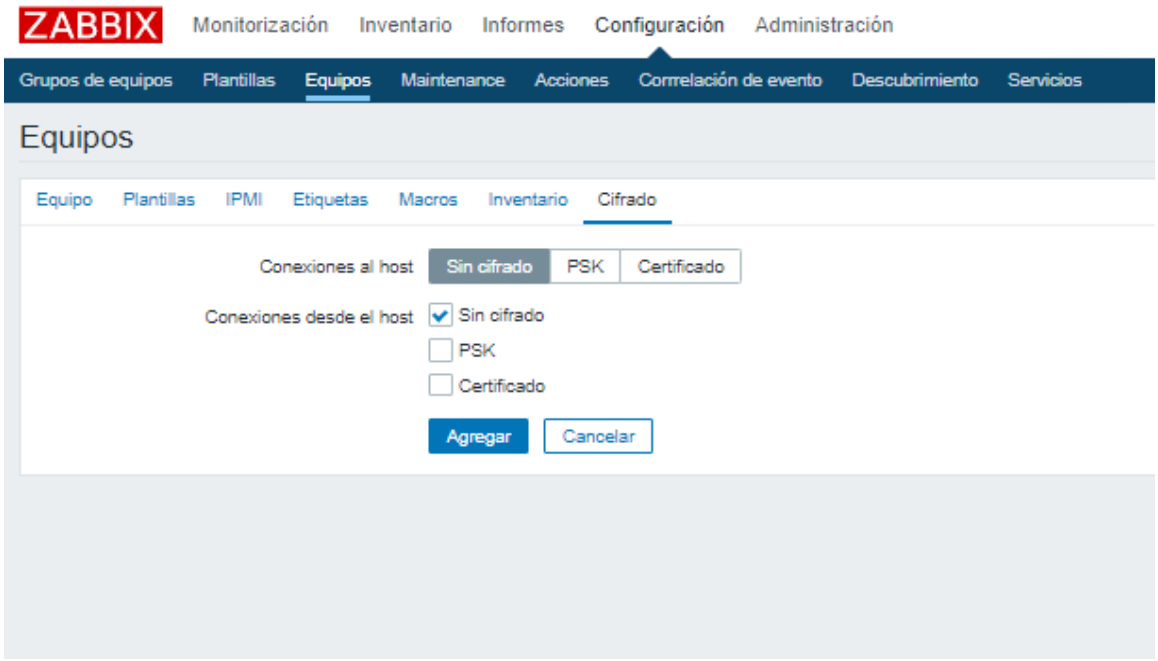


Ilustración 46: Interfaz web de Zabbix - Página de creación de un nuevo equipo III

En esta página, se seleccionó el cifrado PSK para las conexiones al *host*, así como también se seleccionó de forma exclusiva dicho tipo de cifrado para las conexiones desde el *host*. Además, se estableció la identidad de la clave PSK con la especificada en el fichero de configuración del agente del servidor MR, así como el valor de la clave correspondiendo con la generada y almacenada en el fichero */etc/zabbix/zabbix_agentd.psk* del mismo.

Para finalizar la creación del equipo se clicó en el botón "Agregar".

Habilitación de puerto: Puerto 587/tcp

Debido a que luego se configuraría el sistema de monitorización de Zabbix para enviar notificaciones por correo electrónico mediante el protocolo SMTP utilizando el servicio de mensajería de Gmail fue necesario habilitar previamente el puerto 587, mediante una regla, en el cortafuegos del servidor MD a través de *FirewallD*:

```
# firewall-cmd --zone=public --add-port=587/tcp --permanent
# firewall-cmd --reload
```

Configuración de las notificaciones por correo electrónico de Zabbix

Se decidió establecer un método de notificación sobre las posibles alertas del sistema de monitorización de Zabbix basado en mensajes de correo electrónico, de tal forma que estas fueran enviadas a una cuenta de correo específica gestionada por los administradores del centro.

En primer lugar, fue necesario crear una cuenta de correo temporal a la que se enviarían dichas notificaciones.

A continuación, se procedió a configurar el sistema de monitorización para enviar dichas notificaciones a la cuenta de correo creada, en este caso, mediante el servicio de correo electrónico de Gmail.

Para ello, se ingresó de nuevo a la interfaz web de Zabbix y se accedió a *Administración* > *Tipos de Medios*, mostrándose así una página con una lista de los medios de notificación disponibles (Ilustración 47).

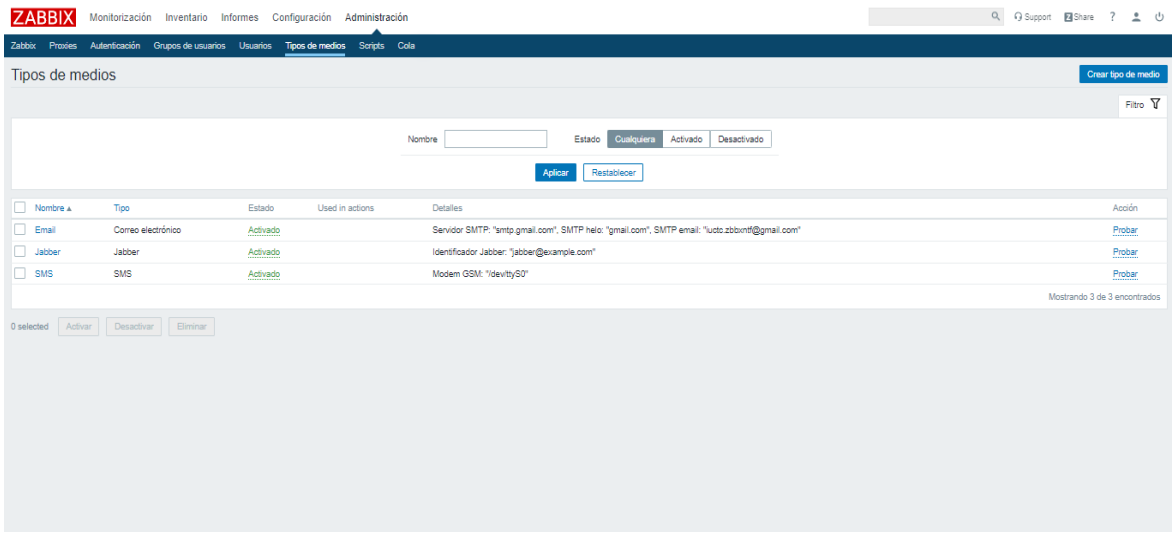


Ilustración 47: Interfaz web de Zabbix - Lista de tipos de medios

En dicha página, se accedió al tipo de medio "Email" y se ajustaron las opciones del protocolo SMTP de acuerdo con la configuración provista por el servicio de mensajería de la cuenta creada, en este caso, Gmail (Ilustración 48).

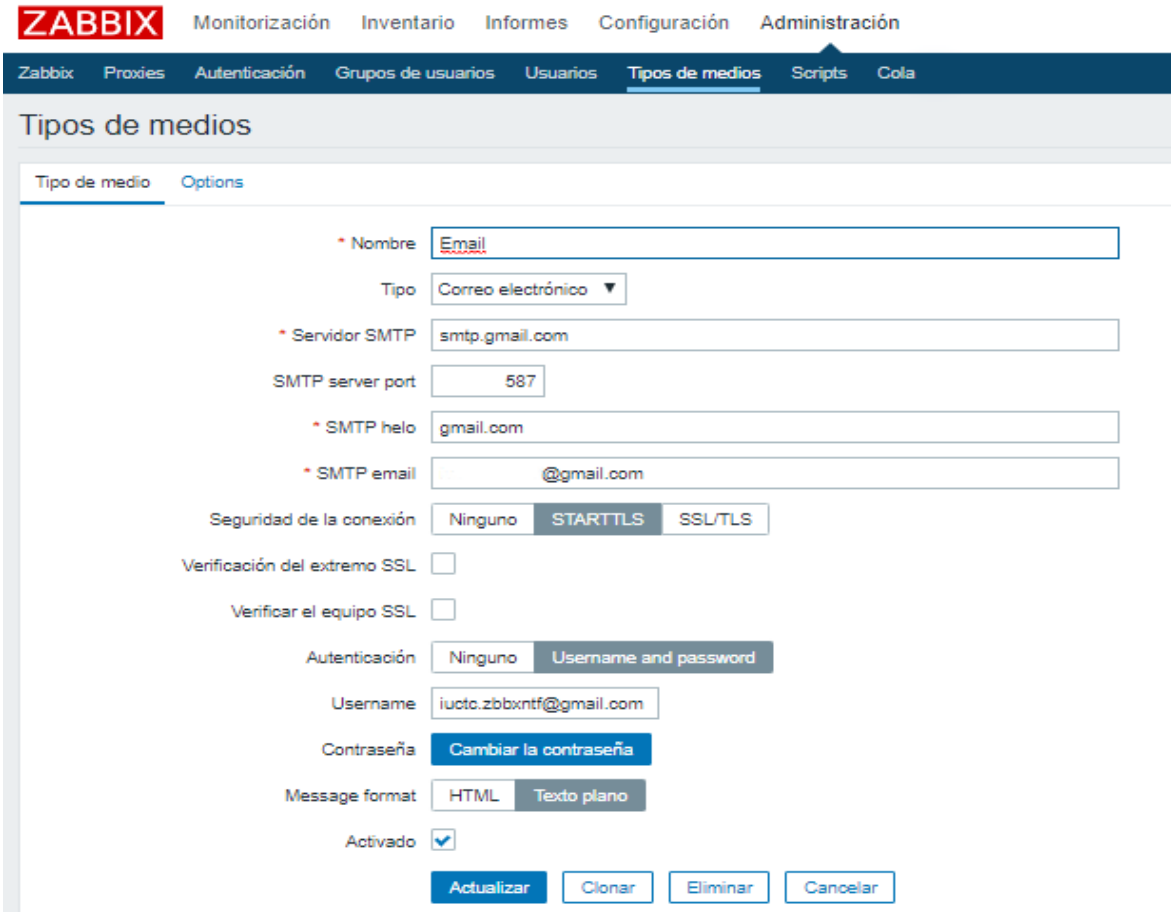


Ilustración 48: Interfaz web de Zabbix - Página de creación de un nuevo tipo de medio

Una vez actualizado el tipo de medio, se creó un nuevo usuario en el sistema de monitorización de Zabbix, para lo cual fue necesario acceder a *Administración* > *Usuarios* (Ilustración 51).

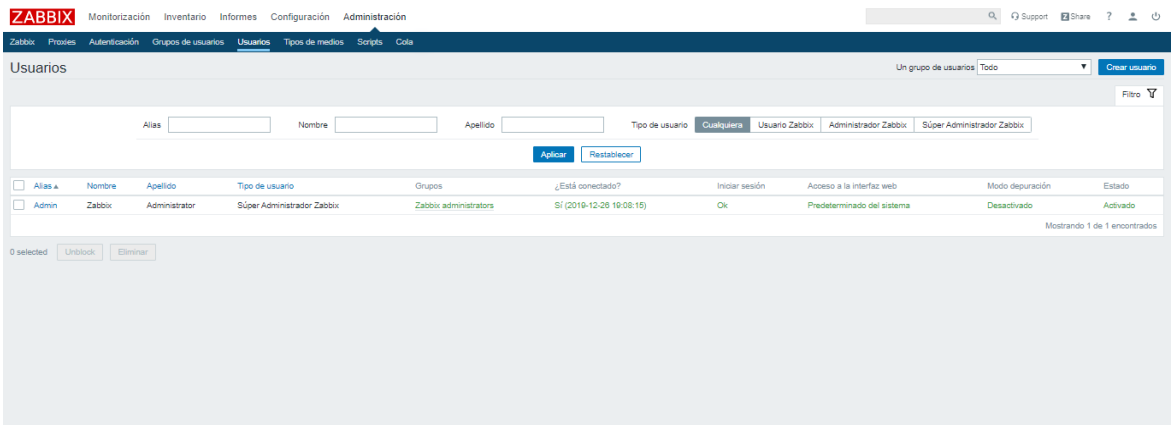


Ilustración 49: Interfaz web de Zabbix - Lista de usuarios

Seguidamente se clicó en el botón "Crear usuario" situado en la esquina superior derecha de la pantalla, mostrándose así la página de creación de un nuevo usuario (Ilustración 50).

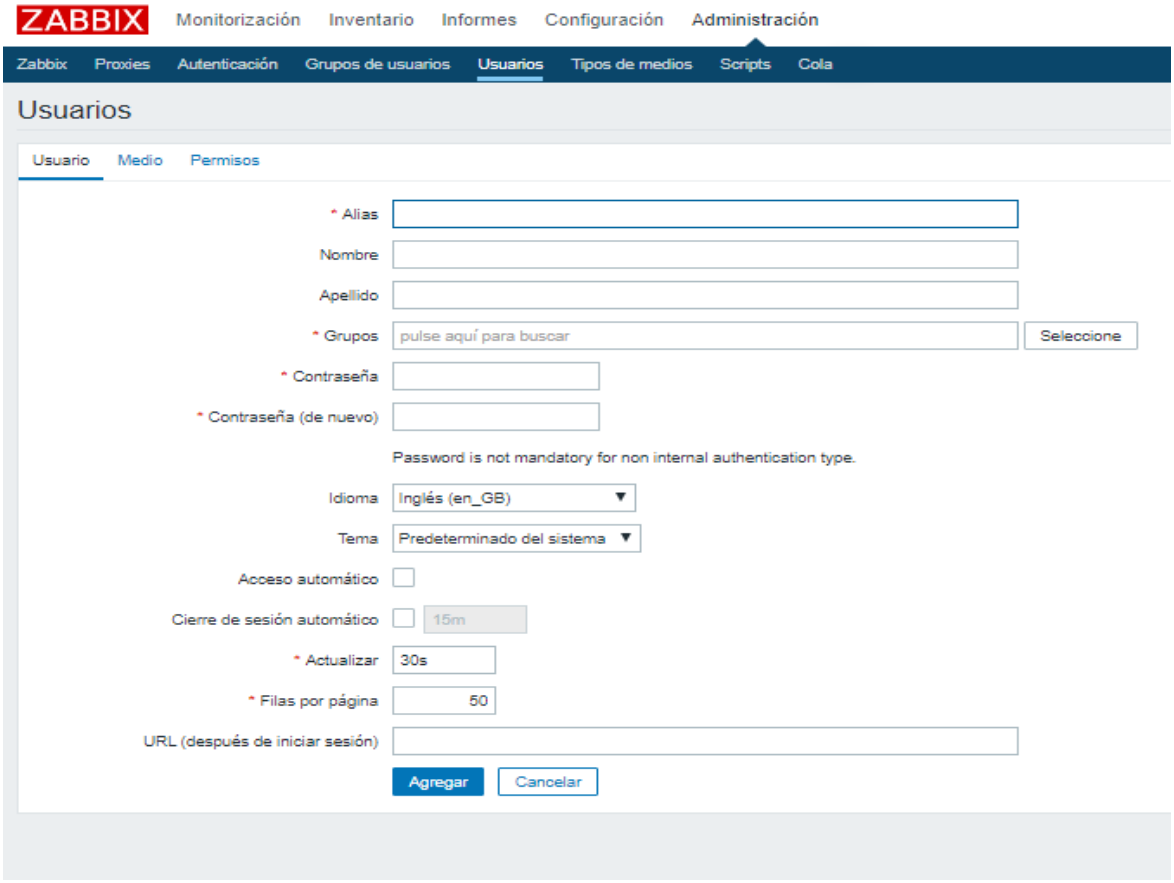


Ilustración 50: Interfaz web de Zabbix - Página de creación de un nuevo usuario I

En dicha página se introdujo el alias para el nuevo usuario, se le estableció una contraseña y se añadió al grupo "Zabbix administrators".

Una vez hecho esto, se accedió a la pestaña "Medio" del propio formulario (Ilustración 51).

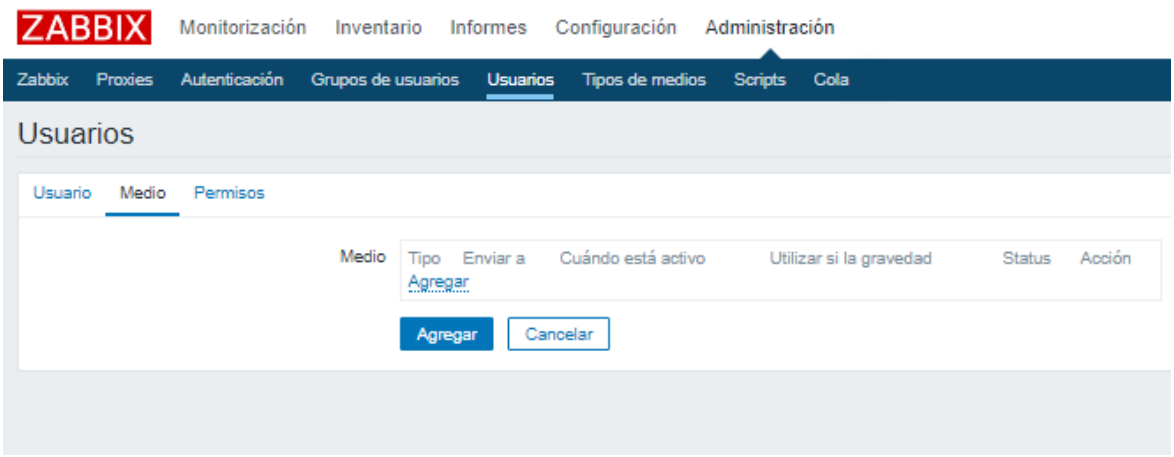


Ilustración 51: Interfaz web de Zabbix - Página de creación de un nuevo usuario II

A continuación, se clicó en el enlace subrayado “Agregar” mostrándose así una ventana emergente con un formulario para añadir un nuevo medio al usuario (Ilustración 52).

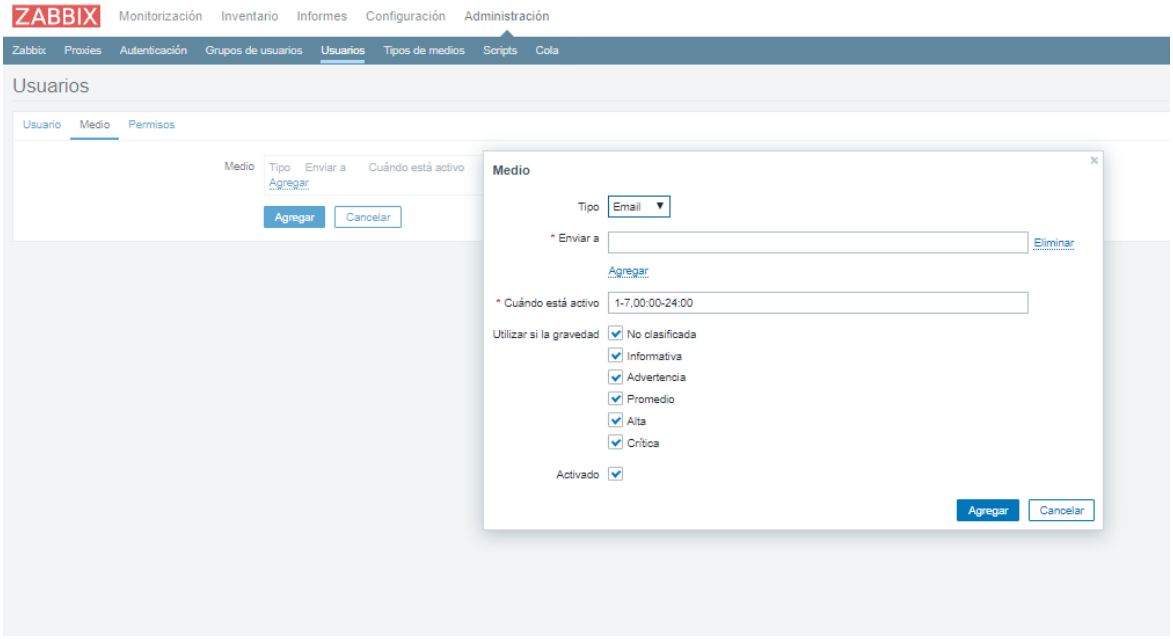


Ilustración 52: Interfaz web de Zabbix - Página de creación de un nuevo usuario III

En dicho formulario se introdujo en el campo “Enviar a” el correo electrónico creado para las notificaciones de Zabbix, manteniéndose los demás campos con los valores mostrados por defecto, y se clicó en el botón “Agregar” situado en la esquina inferior derecha de la ventana emergente.

Para finalizar la creación del usuario se clicó en el botón “Agregar”.

Por último, fue necesario habilitar las notificaciones, para lo cual fue necesario acceder a *Configuración* > *Acciones*, mostrándose así una lista de las acciones de Zabbix (Ilustración 53).

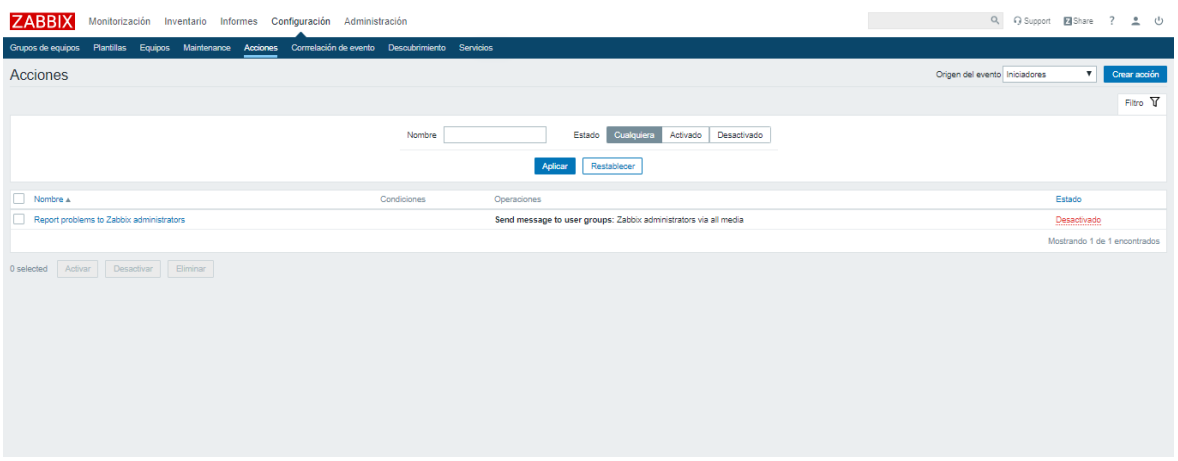


Ilustración 53: Interfaz web de Zabbix - Lista de acciones

En dicha lista se mostraba la acción de notificación de problemas a los usuarios administradores de Zabbix, la cual estaba inicialmente desactivada, por lo que fue necesario habilitarla en la columna "Estado".

Configuración para la monitorización de un servicio web mediante Zabbix

Además, se configuró el sistema de monitorización de Zabbix del servidor MD para monitorizar el estado de los servicios web del centro.

Para ello se ingresó a la interfaz web de Zabbix y acceder a *Configuración* > *Equipos* y se accedió a su vez al equipo correspondiente al servidor MR (Ilustración 54).

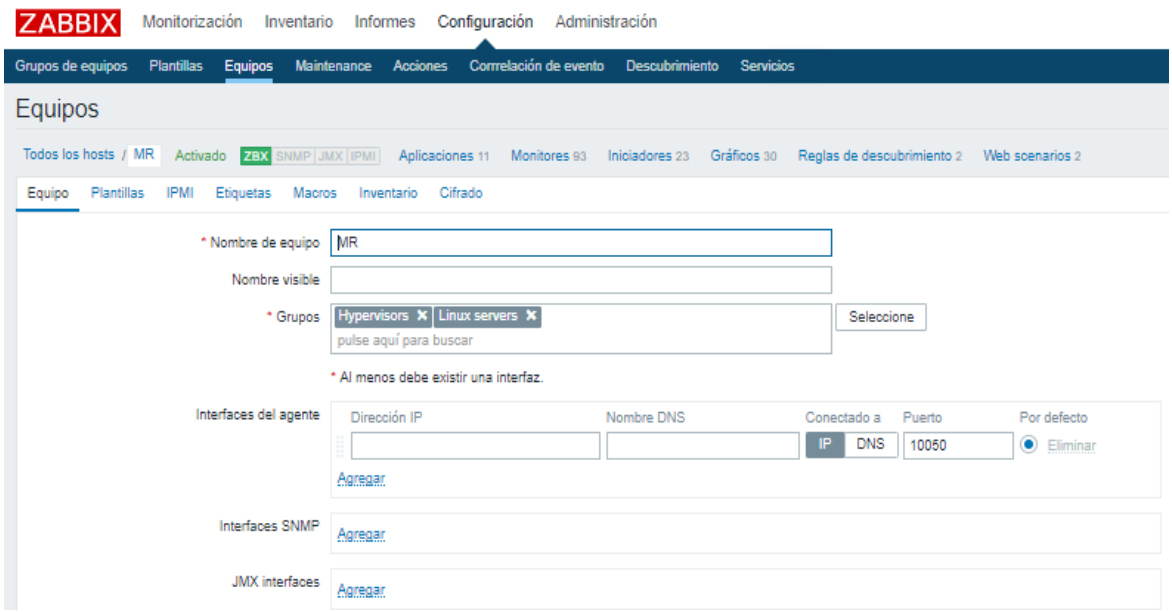


Ilustración 54: Interfaz web de Zabbix - Página de configuración del servidor "MR"

A continuación, se accedió al apartado "Web scenarios" mostrándose así una lista de todos los escenarios web pertenecientes al servidor MR (Ilustración 55).

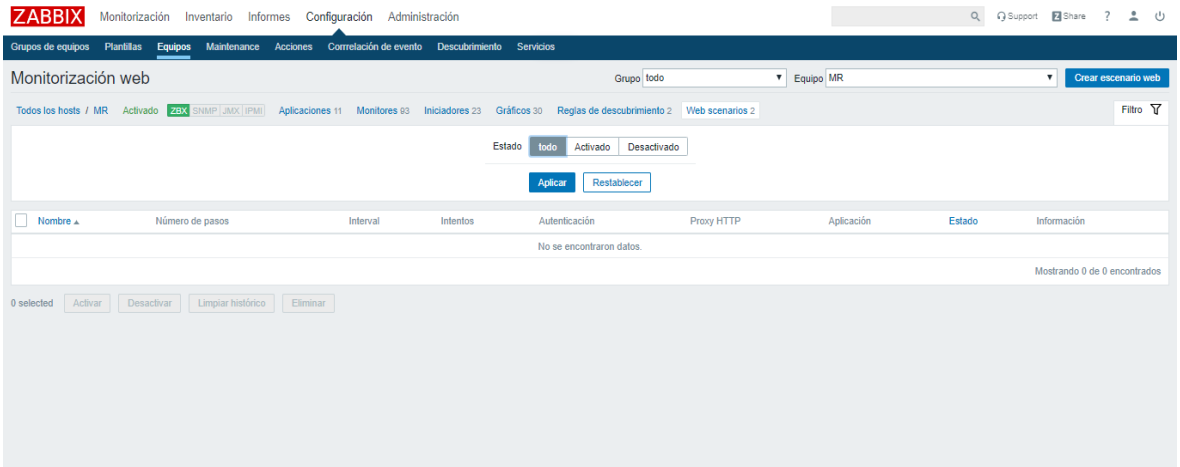


Ilustración 55: Interfaz web de Zabbix - Página de configuración del servidor "MR" - Lista de escenarios web

Para crear un escenario web que monitorizara un servicio web se pulsó el botón “Crear escenario web” situado en la esquina superior derecha de la pantalla, lo que condujo a la página de creación de un nuevo escenario web (Ilustración 56).

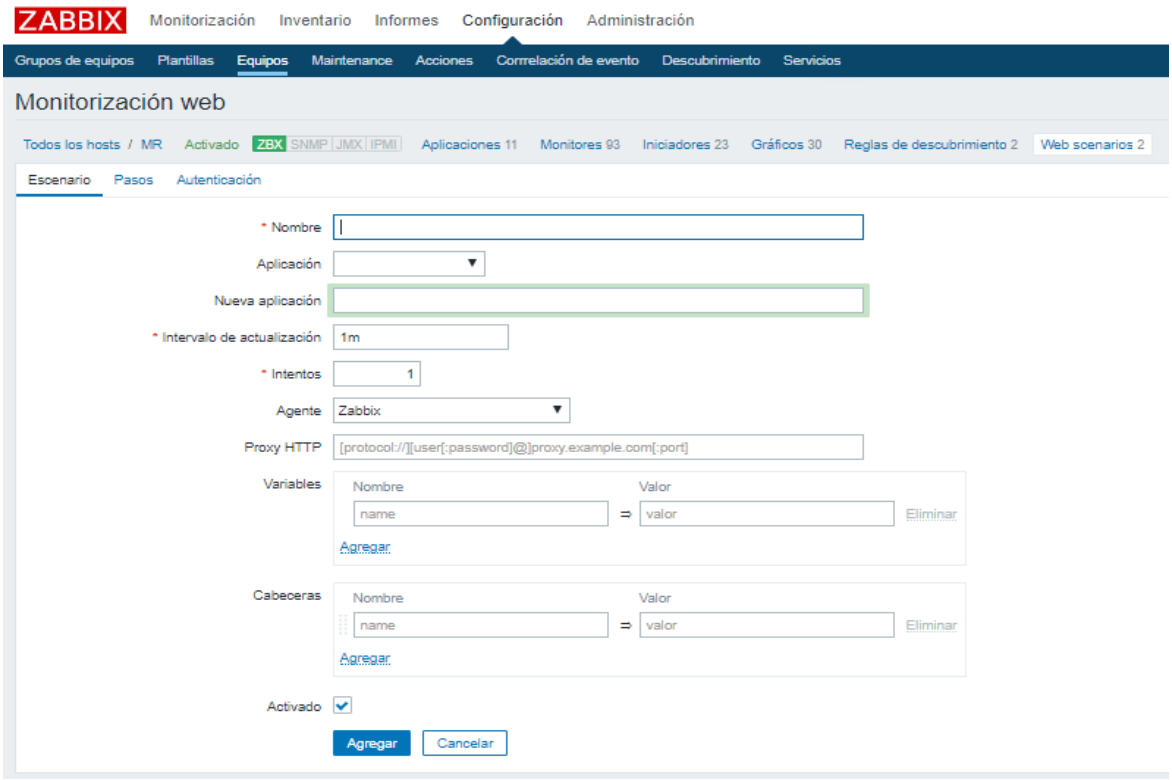


Ilustración 56: Interfaz web de Zabbix - Página de configuración del servidor "MR" - Creación de un nuevo escenario web I

En el apartado “Escenario” se especificó el nombre del escenario web, en este caso un nombre identificativo para el servicio web que se iba a monitorizar, y la aplicación a la que pertenecía dicho escenario, en este caso “Websites”.

A continuación, se accedió en el apartado “Pasos” del mismo formulario (Ilustración 57).

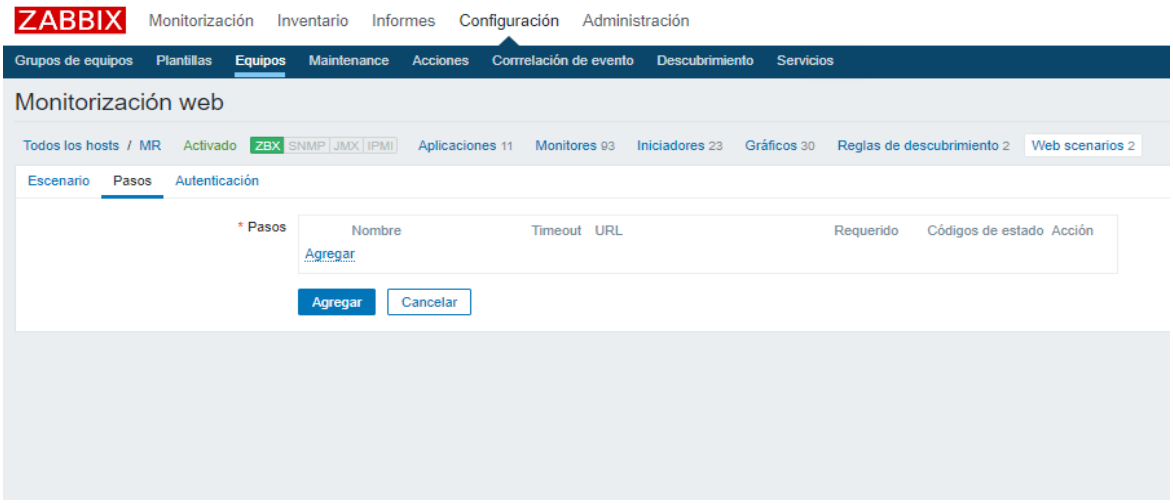


Ilustración 57: Interfaz web de Zabbix - Página de configuración del servidor "MR" - Creación de un nuevo escenario web II

Seguidamente, se clicó en el enlace subrayado “Agregar” mostrándose así una ventana emergente con un formulario para añadir un nuevo paso al escenario web (Ilustración 58).

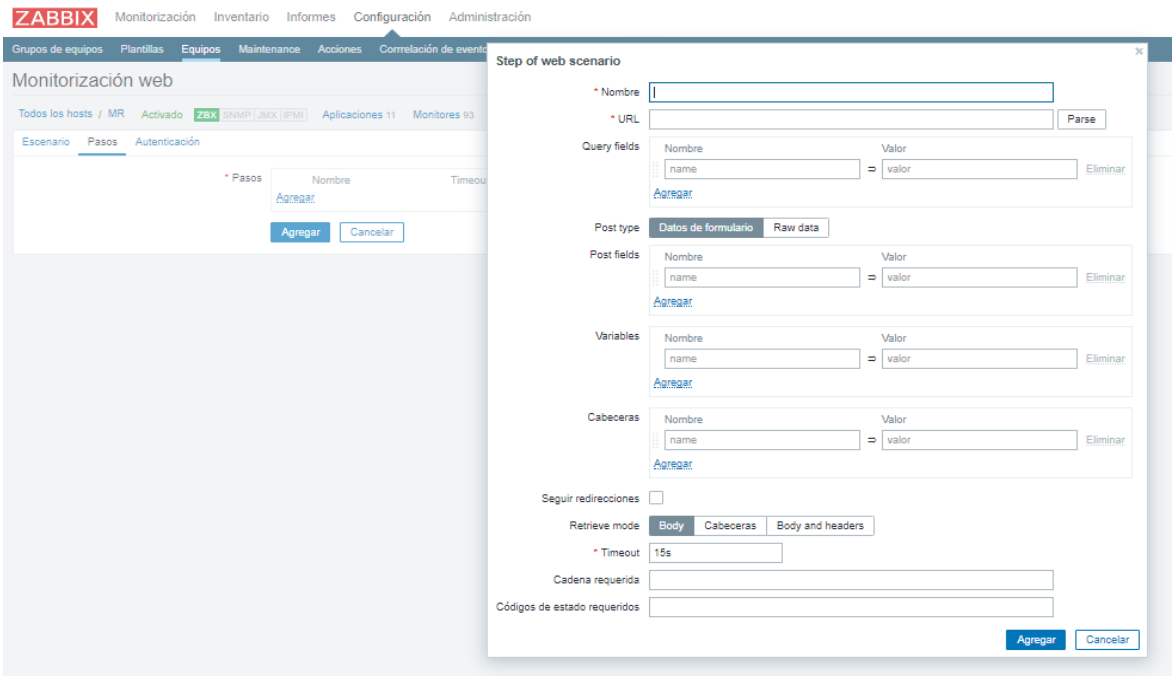


Ilustración 58: Interfaz web de Zabbix - Página de configuración del servidor "MR" - Creación de un nuevo escenario web III

En dicho formulario se especificó el nombre de la página o paso del nuevo escenario web, la URL de la página principal del servicio web a monitorizar y el código de estado HTTP a verificar, en este caso el código HTTP de estado satisfactorio (200), manteniéndose los demás campos con los valores mostrados por defecto, y se clicó en el botón “Agregar” situado en la esquina inferior derecha de la ventana emergente.

Para finalizar la creación del escenario web se clicó en el botón “Agregar”.

Creación de un disparador por cada servicio web monitorizado

El sistema de monitorización Zabbix cuenta con dos piezas de software fundamentales: monitores (*items*) e iniciadores (*triggers*), entre los numerosos componentes que posibilitan su funcionamiento.

Los monitores se encargan de recolectar información sobre los equipos monitorizados, mientras que los iniciadores son expresiones lógicas que permiten comparar dicha información con unos valores límite para establecer si el estado de dicha información es aceptable o no para cada equipo.

Estos elementos vienen incluidos y preconfigurados por defecto con la utilización de las plantillas de Zabbix, sin embargo, como sirven para un propósito general fue necesario crear algunos de estos elementos de forma personalizada para el servidor MR.

En este caso, se requirió la creación de un iniciador por cada servicio web monitorizado que activara una alerta en caso de que su respectivo servicio web estuviera fuera de servicio por más de 5 minutos.

Para crear un iniciador de este tipo, en primer lugar, se accedió a *Configuración* > *Equipos* y se accedió al enlace “Iniciadores” del servidor MR (Ilustración 59).

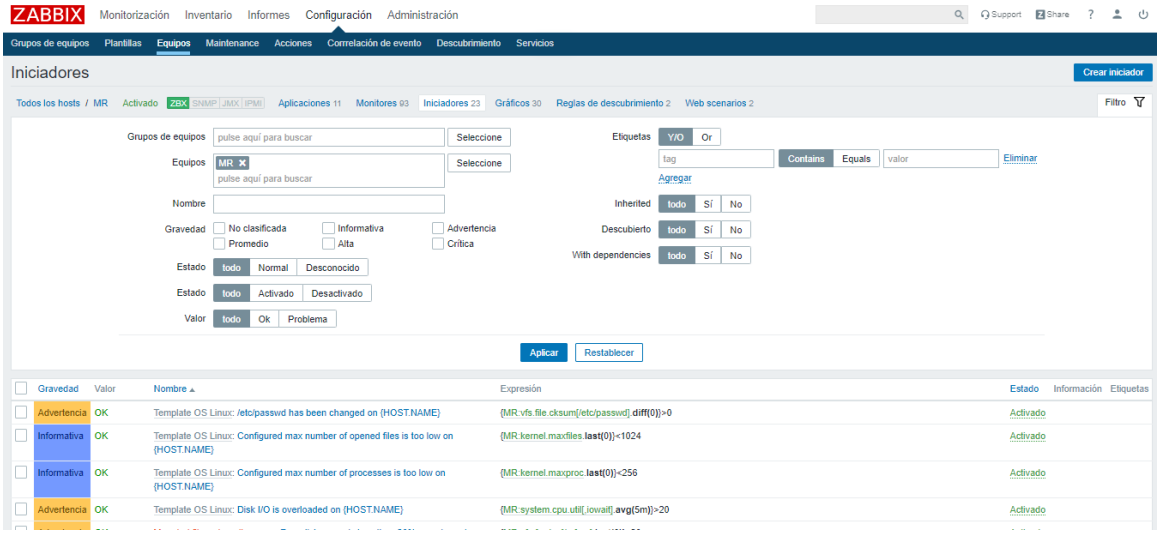


Ilustración 59: Interfaz web de Zabbix - Página de configuración del servidor "MR" - Lista de iniciadores

Seguidamente, se clicó en el botón “Crear iniciador” y se accedió a la página de creación de un nuevo iniciador (Ilustración 60).

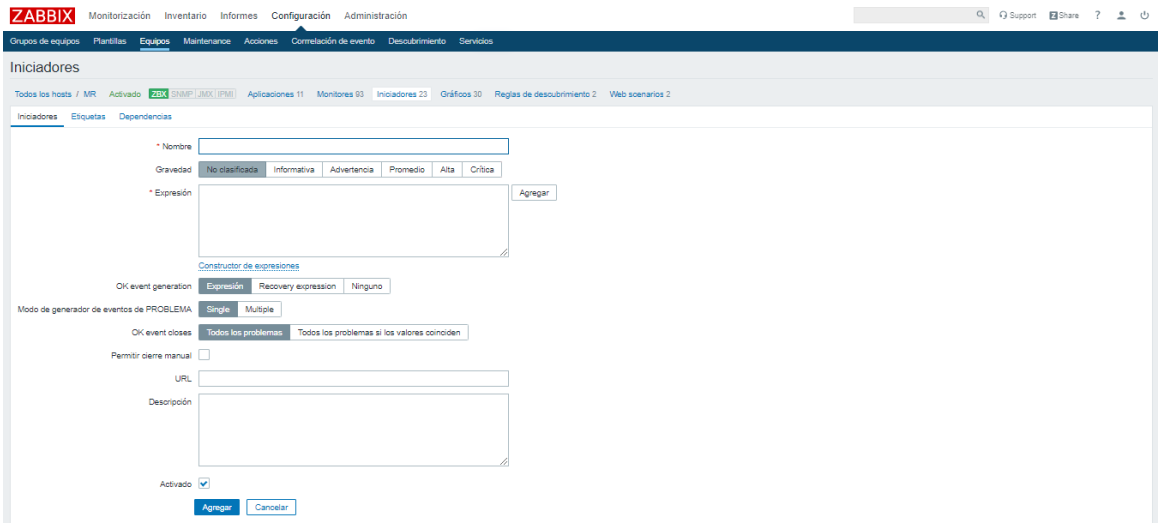


Ilustración 60: Interfaz web de Zabbix - Página de configuración del servidor "MR" - Creación de un nuevo iniciador

La estructura de los iniciadores creados fue la siguiente:

- Nombre. El nombre con el que se mostraría el iniciador al activarse sería en cada caso el siguiente: *Una o varias de las páginas del servicio web* ***NOMBRE_SERVICIO_WEB*** *no responde desde hace más de 5 minutos.*
- Gravedad. La gravedad con la que se activaría el iniciador sería *Alta*.
- Expresión. La expresión lógica que evaluaría el iniciador sería en cada caso la siguiente:
  *{MR:web.test.fail[**NOMBRE_ESCENARIO_WEB**].min(6m)}>0 and*
  *({MR:web.test.fail[**NOMBRE_ESCENARIO_WEB**].time()}<030000 or*
  *{MR:web.test.fail[**NOMBRE_ESCENARIO_WEB**].time()}>033000)*

El desglose de la expresión utilizada en cada iniciador es el siguiente:

- *web.test.fail[**NOMBRE_ESCENARIO_WEB**]*: Monitor que muestra el número del paso fallido de un escenario web, en este caso, ***NOMBRE_ESCENARIO_WEB***. En caso de que todos los pasos sean ejecutados exitosamente mostrará el número 0.
- *min(6m)*: Función que obtiene el valor mínimo de un monitor dado, en este caso *web.test.fail*, dentro de tiempo de evaluación definido, en este caso 6 minutos.
- *time()*: Función que obtiene la hora actual en formato HHMMSS de un monitor, en este caso *web.test.fail*.

De esta manera, se puede comprender que la expresión activará el iniciador cuando el valor mínimo de todos los valores obtenidos en los últimos seis minutos del monitor *web.test.fail* de un determinado escenario web sea diferente de 0, significando que al menos un paso del servicio web monitorizado en dicho escenario web ha quedado fuera de servicio durante ese tiempo:

*{MR:web.test.fail[**NOMBRE_ESCENARIO_WEB**].min(6m)}>0*

Además, esta expresión no se evaluará entre las 03:00 y las 03:30 horas del día, ya que se había programado para ese periodo el proceso de copias de seguridad de los servidores virtuales y se contempla que los servicios web monitorizados puedan estar indisponibles por más de 5 minutos:

*{MR:web.test.fail[**NOMBRE_ESCENARIO_WEB**].time()}<030000 or*
*{MR:web.test.fail[**NOMBRE_ESCENARIO_WEB**].time()}>033000*

Por último, para finalizar la creación del iniciador se clicó en el botón “Agregar”.

Toda la información utilizada relativa al funcionamiento de Zabbix fue consultada en la documentación de su página oficial [46].

ANEXO XXII: INSTALACIÓN DE POWERSHIELD[3]

Se accedió como administrador a cada uno de los servidores físicos de la nueva infraestructura y se descargó de la página oficial de Riello el software mencionado, concretamente llamado PowerShield[3]:

```
# wget RUTA_WEB_POWERSHIELD3
```

Debido a que el fichero descargado era de tipo RPM fue necesario instalarlo con la herramienta adecuada:

```
# rpm -i FICHERO_POWERSHIELD3.rpm
```

Esto generó en el directorio */opt/upsmon/* todos los ficheros necesarios para la instalación, configuración y gestión de los sistemas de alimentación ininterrumpida de Riello.

Para comenzar la instalación se accedió a dicho directorio:

```
# cd /opt/upsmon/
```

Y se ejecutó el *script* de instalación desde la interfaz de línea de comandos:

```
# ./upsetup
```

Después de elegir el idioma de instalación, el instalador requirió el código PRTK (código especial de los UPS de Riello) del UPS conectado al servidor en cuestión. Tras introducirlo, se mostró el menú de configuración de dicho sistema (Ilustración 61).

```
*************************************************************************
***                        Configuración                             ****
***           UPSetup v 6.0.7 04/2018 Copyright 2018                  ****
***                                                                    ****
***                        Configuración                             ****
*************************************************************************
     1 - Parámetros generales
 --> 2 - Configuración SAI
     3 - Configuración de mensajes
     4 - Planificador acciones
     5 - Configuración de trabajo
-------------------------------------------------------------------------
     0 - Salir

*************************************************************************
```

Ilustración 61: Menú de configuración de UPS de PowerShield[3]

ANEXO XXIII: CONFIGURACIÓN DE UPS A TRAVÉS DE POWERSHIELD[3]

Para configurar el UPS fue necesario acceder al apartado "Configuración SAI", el cual mostró una lista de todas las conexiones UPS definidas.

En dicha lista se definió la nueva conexión especificando:

1) El nombre deseado para el UPS.
2) El código PRTK del UPS.
3) El número de serie del UPS.
4) La conexión del dispositivo. Para esto fue necesario especificar además el tipo de conexión, en este caso local, el puerto serie y su dirección.
5) La habilitación del apagado controlado del servidor conectado al UPS una vez transcurridos, en este caso, 20 minutos desde el corte del suministro eléctrico (traducido como falta de red).
6) La habilitación del apagado controlado del servidor conectado al UPS al descender el tiempo restante de batería del UPS por debajo de, en este caso, 5 minutos (traducido como autonomía).
7) La habilitación del apagado del UPS una vez transcurridos, en este caso, 5 segundos desde el apagado controlado completo del servidor conectado a este.

Esta configuración se utilizó indistintamente para los servidores NG, MR y MD (Ilustración 62).

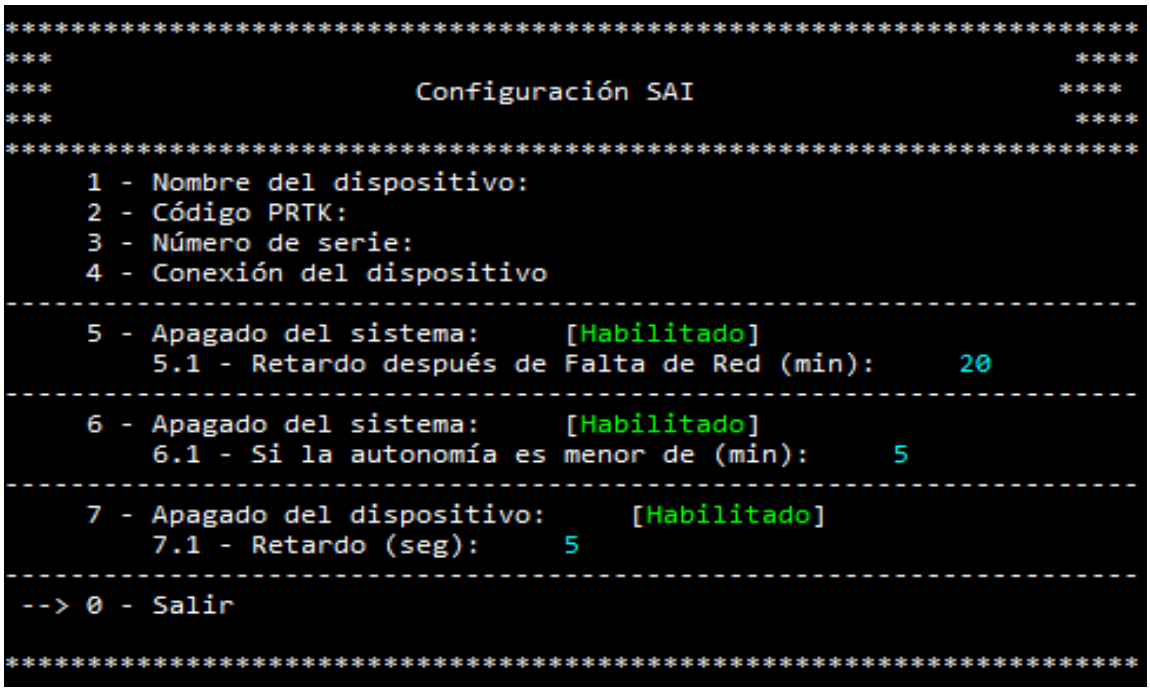


Ilustración 62: Configuración personalizada de UPS mediante PowerShield[3]

Tras esto se salió del menú de configuración proporcionado por PowerShield[3] guardando los cambios realizados.

Además, fue necesario asegurarse de que la opción “AC Recovery” (encendido automático tras la restauración del suministro eléctrico) estuviera habilitado en la BIOS de los servidores NG, MR y MD para que los sistemas de alimentación ininterrumpida pudieran cumplir con su función.

ANEXO XXIV: COMPROBACIÓN DE FUNCIONAMIENTO DEL *SCRIPT VMBACKUP.SH* DEL SISTEMA DE COPIAS DE SEGURIDAD IMPLEMENTADO

En primer lugar, se ejecutó de forma manual y en el momento adecuado, a modo de prueba, dicho *script*:

```
# /usr/local/bin/vmbackuptools/vmbackup.sh
```

Una vez finalizada la ejecución del *script*, se montó la carpeta compartida de almacenamiento de las copias de seguridad "CopiasMV" ubicada en el servidor de copias de seguridad B2 en el directorio */mnt/vmbackups/* del servidor MR:

```
# mount -t cifs //DIRECCIÓN_IP_SERVIDOR_B2/CopiasMV/ -o user=NOMBRE_USUARIO_SERVIDOR_B2,pass=CONTRASEÑA_USUARIO_SERVIDOR_B2,vers=1.0 /mnt/vmbackups/
```

A continuación, se accedió al directorio de montaje de la carpeta compartida:

```
# cd /mnt/vmbackups/
```

Seguidamente, se accedió al directorio en el cual se almacenaron las copias de seguridad realizadas por el *script* ejecutado, comprobando que tuviera la estructura correspondiente a la fecha de ejecución del susodicho:

```
# cd AAAA/MM/DD/
```

Luego se listaron los ficheros de dicho directorio y se obtuvo un conjunto de directorios correspondientes a las copias de seguridad de cada servidor virtual. Con esto se pudo comprobar que el número y el nombre de estos correspondieran al número y el nombre de los servidores virtuales copiados en la ejecución del *script* de copias de seguridad.

Tras ello, se accedió al directorio de cada copia de seguridad:

```
# cd NOMBRE_SERVIDOR_VIRTUAL/
```

Inmediatamente después, se listó el contenido de cada directorio y se comprobó la existencia de, al menos, dos ficheros:

- Un fichero de tipo XML (.xml) correspondiente a la copia del fichero de configuración del servidor virtual en cuestión: ***NOMBRE_SERVIDOR_VIRTUAL****.xml*.
- Al menos un fichero comprimido de tipo QCOW2 (.qcow2.gz) correspondiente a la copia de la imagen o imágenes de disco del servidor virtual en cuestión: ***NOMBRE_IMAGEN_DISCO_SERVIDOR_VIRTUAL****.qcow2.gz.*

Una vez realizadas las comprobaciones pertinentes, se retornó al directorio */mnt* y se desmontó la carpeta compartida de almacenamiento de copias de seguridad del directorio de montaje del servidor MR:

```
# cd /mnt/
# umount /mnt/vmbackups/
```

Además, se verificó de la misma manera el funcionamiento del *script* ejecutado mediante Cron, asegurando además que dicho proceso se realizara en el periodo programado.

Por último, se comprobó que todos los eventos llevados a cabo por el *script* fueran registrados en el fichero *vmbackup.log*:

```
# cat /usr/local/bin/vmbackuptools/vmbackup.log
```

ANEXO XXV: COMPROBACIÓN DE FUNCIONAMIENTO DEL *SCRIPT VMBACKUP_CLEAN.SH* DEL SISTEMA DE COPIAS DE SEGURIDAD IMPLEMENTADO

Se realizó una pequeña prueba consistente en crear un directorio en la carpeta compartida de almacenamiento de las copias de seguridad "CopiasMV" del servidor B2 que pudiera ser eliminado por el *script* de limpieza de copias de seguridad.

En primer lugar, se montó la carpeta compartida de almacenamiento de las copias de seguridad "CopiasMV" del servidor B2 en el directorio */mnt/vmbackups/* del servidor MR:

```
# mount -t cifs //DIRECCIÓN_IP_SERVIDOR_B2/CopiasMV/ -o user=NOMBRE_USUARIO_SERVIDOR_B2,pass=CONTRASEÑA_USUARIO_SERVIDOR_B2,vers=1.0 /mnt/vmbackups/
```

A continuación, se accedió al directorio de montaje de la carpeta compartida:

```
# cd /mnt/vmbackups/
```

Seguidamente, se creó un directorio con el formato adecuado para ser eliminado por el *script* de limpieza de copias de seguridad:

```
# mkdir AAAA/MM/DD/
```

Además, se crearon algunos ficheros en dicho directorio para comprobar que su contenido también fuera eliminado al ejecutar el *script*.

Después se retornó al directorio */mnt* y se desmontó la carpeta compartida de almacenamiento de copias de seguridad del directorio de montaje del servidor MR:

```
# cd /mnt/
# umount /mnt/vmbackups/
```

Tras ello, se ejecutó de forma manual el *script* de limpieza de copias de seguridad:

```
# /usr/local/bin/vmbackuptools/vmbackup_clean.sh
```

Una vez ejecutado el *script*, se volvió a montar la carpeta compartida de almacenamiento de las copias de seguridad "CopiasMV" del servidor B2 en el directorio */mnt/vmbackups/* del servidor MR y se accedió al directorio de montaje de la carpeta compartida */mnt/vmbackups/*.

De esta forma se comprobó que el directorio ***AAAA/MM/DD/*** creado anteriormente en la carpeta compartida fue eliminado con éxito junto con todo su contenido.

Tras realizar dicha comprobación, se retornó al directorio */mnt* y se desmontó la carpeta compartida de almacenamiento de copias de seguridad del directorio de montaje del servidor MR.

Además, se verificó de la misma manera el funcionamiento del *script* ejecutado mediante Cron, asegurando además que dicho proceso se realizara en el periodo programado.

Por último, se comprobó que todos los eventos llevados a cabo por el *script* fueran registrados en el fichero *vmbackup.log*:

```
# cat /usr/local/bin/vmbackuptools/vmbackup.log
```

ANEXO XXVI: COMPROBACIÓN DE FUNCIONAMIENTO DEL *SCRIPT VMRESTORE.SH* DEL SISTEMA DE COPIAS DE SEGURIDAD IMPLEMENTADO

En primer lugar, fue prerrequisito que existieran copias de seguridad en la carpeta compartida de almacenamiento "CopiasMV" del servidor B2, concretamente de la máquina virtual de prueba *NOMBRE_MV*, para así poder proceder con su restauración.

Seguidamente, se ejecutó el *script* de restauración de copias de seguridad:

```
# /usr/local/bin/vmbackuptools/vmrestore.sh
```

Una vez iniciado el *script* se mostraron algunos mensajes informativos y se requirió la entrada de datos del usuario permitiendo así la introducción de la ruta del fichero XML del servidor virtual a restaurar de la carpeta compartida de copias de seguridad:

```
### BIENVENIDO AL SCRIPT DE RESTAURACIÓN DE DOMINIOS DE KVM (PARA SALIR PULSE CTRL-C Ó CTRL-Z) ###
Servidor de copias de seguridad montado.
Introduzca la ruta absoluta del fichero XML de la copia del dominio a restaurar: /mnt/vmbackups/
```

En este punto se comprobó que el *script* verificara correctamente las siguientes condiciones al introducir una ruta de fichero:

- Tratarse de un fichero que existiera.
- Tratarse de un fichero regular.
- Tratarse de un fichero de tipo XML, es decir, con extensión XML.

En caso de incumplirse alguna de las condiciones anteriores el *script* volvería a requerir una ruta de fichero.

Tras introducir una ruta de fichero válida, en este caso, el fichero XML de la copia de seguridad de la máquina virtual de prueba *NOMBRE_MV* se mostró una confirmación de la operación antes de proceder con la restauración:

```
¿Está seguro de que quiere restaurar el dominio NOMBRE_SERVIDOR_VIRTUAL a partir del fichero /mnt/vmbackups/AAAA/MM/DD/ NOMBRE_SERVIDOR_VIRTUAL / NOMBRE_SERVIDOR_VIRTUAL.xml? [S/N]
```

En este punto se comprobó que el *script* verificara correctamente las siguientes condiciones al introducir una respuesta:

- Que la respuesta introducida sea "S", en cuyo caso continuaría con la siguiente línea de ejecución.
- Que la respuesta introducida sea "N", en cuyo caso el *script* volvería a requerir una ruta de fichero.
- Que la respuesta no sea ninguna de las anteriores, en cuyo caso se volvería a requerir la confirmación de restauración a partir de la ruta de fichero anteriormente especificada.

Tras confirmar la restauración, en este caso, de la máquina virtual de prueba *NOMBRE_MV* se procedió con esta.

Una vez realizada la restauración de la máquina virtual *NOMBRE_MV* se comprobó que esta fuera realizada correctamente verificando que:

- En el directorio de almacenamiento de los ficheros XML de los dominios de KVM */etc/libvirt/qemu/* existiera el fichero XML restaurado correspondiente a la máquina virtual *NOMBRE_MV*.
- En el directorio de almacenamiento de las imágenes de disco de los dominios de KVM */var/lib/libvirt/images/* existiera la imagen o imágenes de disco restaurados correspondientes a la máquina virtual *NOMBRE_MV*.

Tras realizar las comprobaciones pertinentes, se eliminó la máquina virtual de prueba *NOMBRE_MV*.

ANEXO XXVII: COMPROBACIÓN DEL FUNCIONAMIENTO DE LA MONITORIZACIÓN DE EQUIPOS DE ZABBIX

Se ingresó a la interfaz web de Zabbix y se accedió a *Configuración* > *Equipos* para comprobar que tanto la disponibilidad como el método de cifrado de cada equipo monitorizado estuvieran en orden (Ilustración 63).

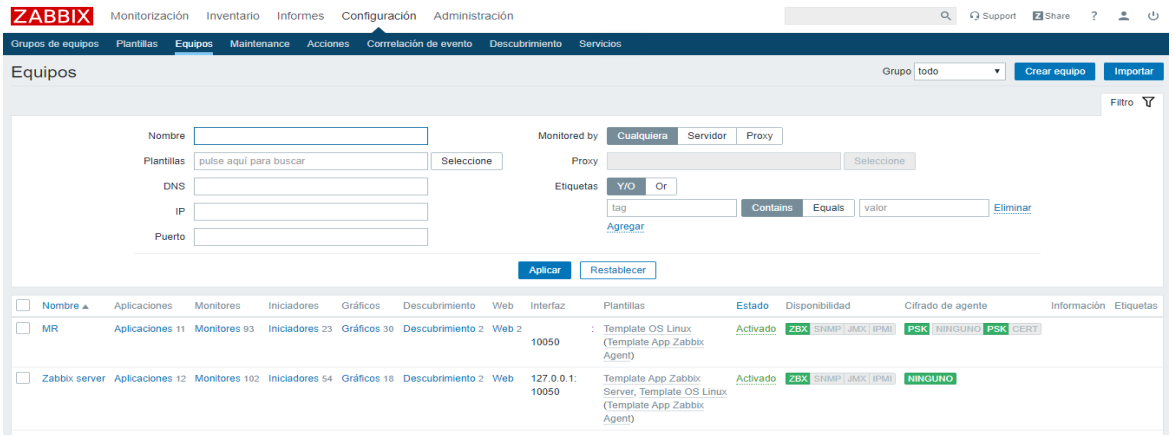


Ilustración 63: Interfaz web de Zabbix - Verificación de estado de los equipos monitorizados

Además, se hizo una pequeña prueba consistente en realizar una acción en el servidor MR que activase uno de los iniciadores de este en el sistema de monitorización para así comprobar que estuviera siendo monitorizado correctamente.

Para ello, ya que no se pretendía realizar acciones que comprometieran el funcionamiento de la infraestructura, simplemente se decidió deshabilitar el servicio del agente de Zabbix en el servidor MR accediendo como administrador a este y ejecutando el siguiente comando:

```
# systemctl stop zabbix-agent
```

De manera que, transcurridos unos minutos, el sistema de monitorización detectara un problema de comunicación con el servidor MR (Ilustración 64).

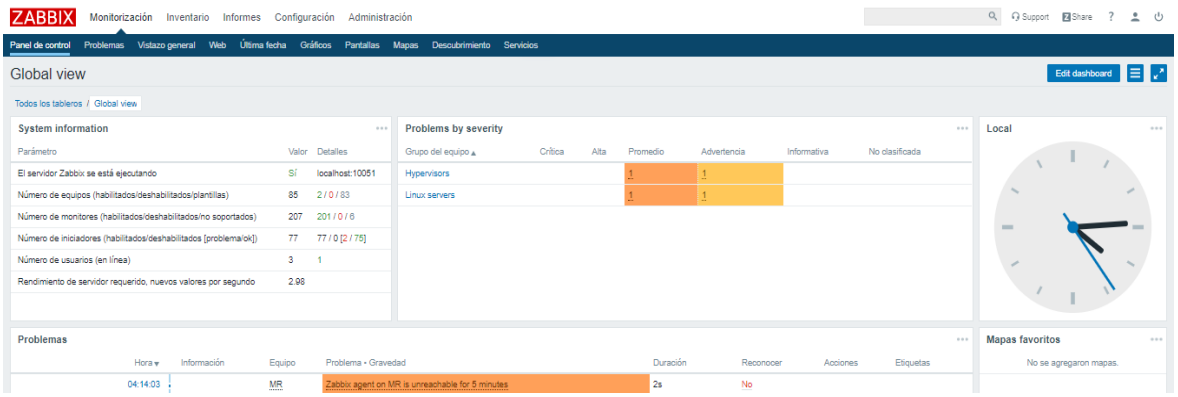


Ilustración 64: Interfaz web de Zabbix - Página de inicio - Verificación de problema de comunicación con un equipo

Seguidamente, se comprobó que este problema era notificado al correo electrónico dispuesto para Zabbix (Ilustración 65).

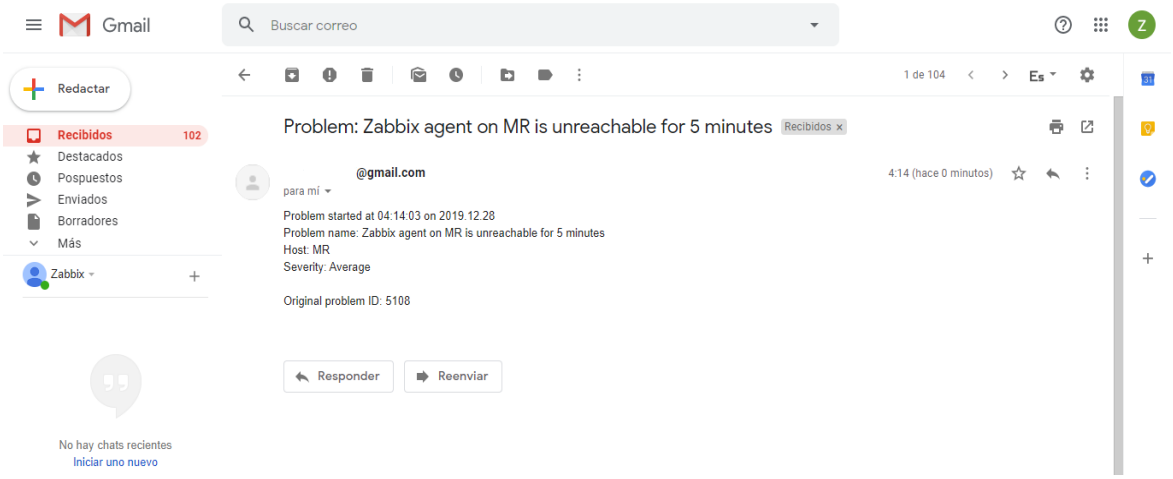


Ilustración 65: Notificación de Zabbix por correo electrónico de un problema de comunicación con un equipo

A continuación, se volvió a habilitar el servicio del agente de Zabbix en el servidor MR:

```
# systemctl start zabbix-agent
```

Por último, se comprobó que la resolución del problema era notificada de la misma forma al correo electrónico dispuesto para Zabbix (Ilustración 66).

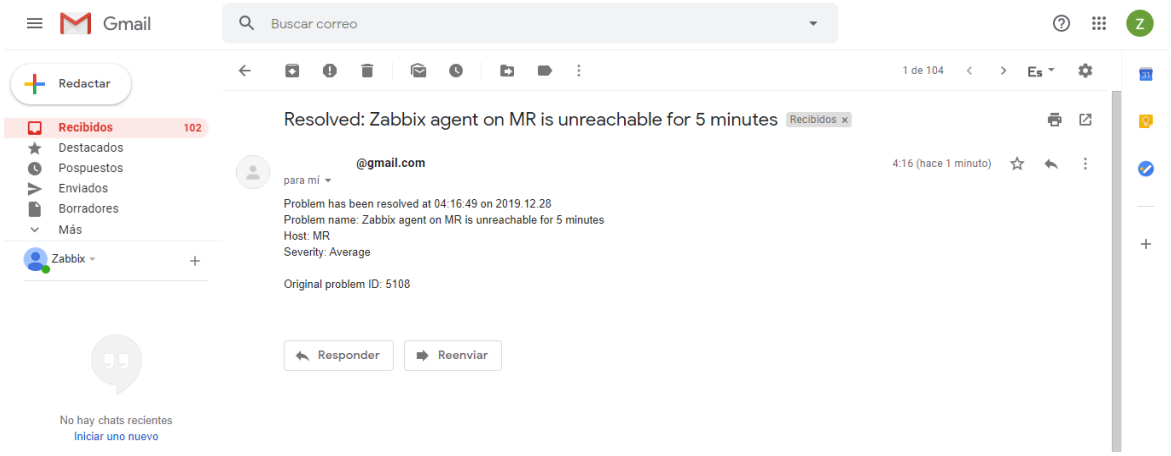


Ilustración 66: Notificación de Zabbix por correo electrónico de una resolución de problema de comunicación con un equipo

ANEXO XXVIII: COMPROBACIÓN DEL FUNCIONAMIENTO DE LA MONITORIZACIÓN DE SERVICIOS WEB DE ZABBIX

Se ingresó a la interfaz web de Zabbix y se accedió a *Monitorización* > *Web* para comprobar que la disponibilidad de los servicios web monitorizados estaba en orden (Ilustración 67).

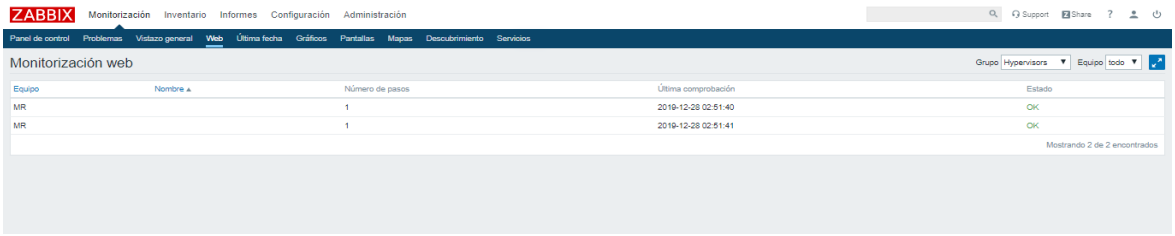


Ilustración 67: Interfaz web de Zabbix - Página de configuración del servidor "MR" - Verificación de estado de los escenarios web

Además, se hizo una pequeña prueba consistente en dejar fuera de servicio uno de los servicios web monitorizados y así se activase su iniciador correspondiente en el sistema de monitorización para así comprobar que estuviera siendo monitorizado correctamente.

Para ello, se decidió apagar brevemente el servidor virtual que alojaba uno de dichos servicios web, de forma que este quedara momentáneamente indisponible y el sistema de monitorización detectara el problema (Ilustración 68).

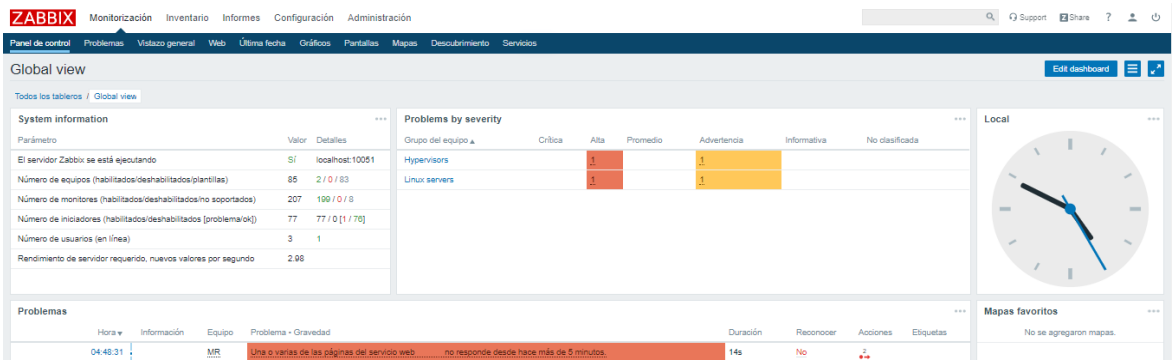


Ilustración 68: Interfaz web de Zabbix - Página de inicio - Verificación de problema de disponibilidad de un servicio web

Seguidamente, se comprobó que este problema era notificado al correo electrónico dispuesto para Zabbix (Ilustración 69).

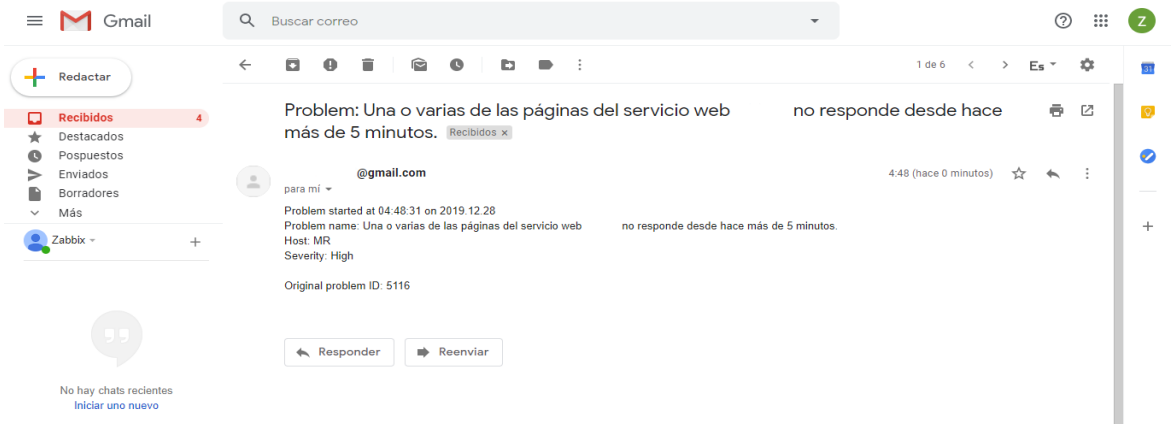


Ilustración 69: Notificación de Zabbix por correo electrónico de un problema de disponibilidad de un servicio web

A continuación, se volvió a iniciar el servidor virtual apagado anteriormente, de manera que el servicio web deshabilitado quedara nuevamente disponible.

Por último, se comprobó que la resolución del problema era notificada de la misma forma al correo electrónico dispuesto para Zabbix (Ilustración 70).

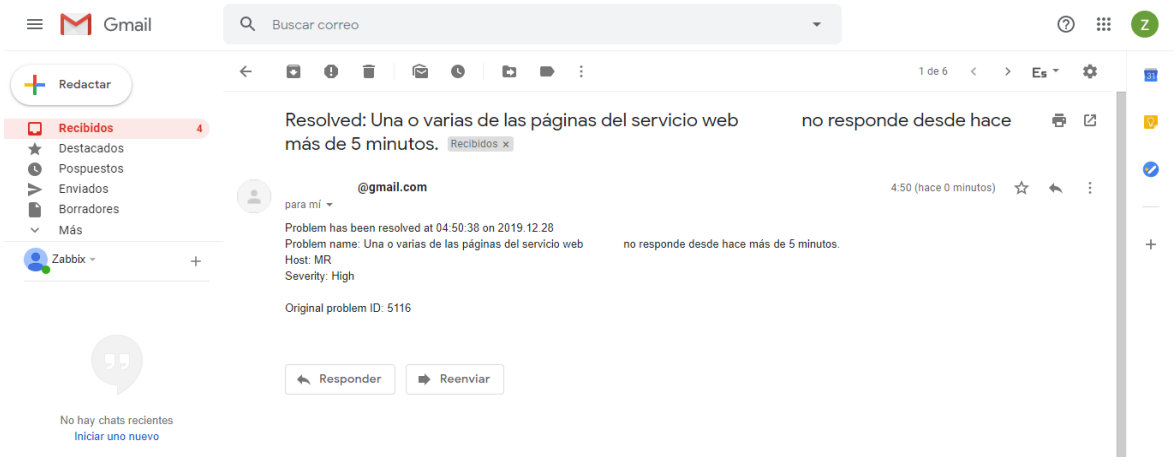


Ilustración 70: Notificación de Zabbix por correo electrónico de una resolución de problema de disponibilidad de un servicio web

FUENTES DE INFORMACIÓN

[1] IUCTC, "Presentación y objetivos del IUCTC", (en línea; Junio de 2020). Disponible en:

http://iuctc.ciber.ulpgc.es/sobre-el-iuctc/presentacion/

[2] EII, "Objetivos y competencias del Grado en Ingeniería Informática", (en línea; Junio de 2020). Disponible en:

https://www.eii.ulpgc.es/tb_university_ex/?q=objetivos-y-competencias-del-gii

[3] Wikipedia, "Definición de una Licencia de Software", (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Licencia_de_software

[4] Wikipedia, "Descripción de la licencia de software GNU GPL", (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/GNU_General_Public_License

[5] GNU, "Términos y condiciones de la licencia de software GNU GPLv1", (en línea; Junio de 2020). Disponible en:

https://www.gnu.org/licenses/old-licenses/gpl-1.0.html

[6] GNU, "Términos y condiciones de la licencia de software GNU GPLv2", (en línea; Junio de 2020). Disponible en:

https://www.gnu.org/licenses/old-licenses/gpl-2.0.html

[7] GNU, "Términos y condiciones de la licencia de software GNU GPLv3", (en línea; Junio de 2020). Disponible en:

https://www.gnu.org/licenses/gpl-3.0.html

[8] Wikipedia, "Descripción de la licencia de software GNU LGPL", (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/GNU_Lesser_General_Public_License

[9] GNU, "Términos y condiciones de la licencia de software GNU LGPLv2", (en línea; Junio de 2020). Disponible en:

https://www.gnu.org/licenses/old-licenses/lgpl-2.0.html

[10] GNU, "Términos y condiciones de la licencia de software GNU LGPLv2.1", (en línea; Junio de 2020). Disponible en:

https://www.gnu.org/licenses/old-licenses/lgpl-2.1.html

[11] GNU, "Términos y condiciones de la licencia de software GNU LGPLv3", (en línea; Junio de 2020). Disponible en:

https://www.gnu.org/licenses/lgpl-3.0.html

[12] Wikipedia, "Descripción de la licencia de software BSD", (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Licencia_BSD

[13] Free Software Directory, "Cláusulas de la licencia de software BSD original", (en línea; Junio de 2020). Disponible en:

https://directory.fsf.org/wiki/License:BSD-4-Clause

[14] Free Software Directory, “Cláusulas de la licencia de software BSD modificada”, (en línea; Junio de 2020). Disponible en:

https://directory.fsf.org/wiki/License:BSD-3-Clause

[15] Free Software Directory, “Cláusulas de la licencia de software BSD simplificada”, (en línea; Junio de 2020). Disponible en:

https://directory.fsf.org/wiki/License:BSD-2-Clause

[16] Wikipedia, “Descripción de la licencia de software Apache”, (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Apache_License

[17] Apache, “Cláusulas de la licencia de software Apache 1.0”, (en línea; Junio de 2020). Disponible en:

https://www.apache.org/licenses/LICENSE-1.0

[18] Apache, “Cláusulas de la licencia de software Apache 1.1”, (en línea; Junio de 2020). Disponible en:

https://www.apache.org/licenses/LICENSE-1.1

[19] Apache, “Cláusulas de la licencia de software Apache 2.0”, (en línea; Junio de 2020). Disponible en:

https://www.apache.org/licenses/LICENSE-2.0

[20] Wikipedia, “Descripción de la licencia de software PHP”, (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Licencia_PHP

[21] Wikipedia, “Términos y condiciones de la versión 3 de la licencia de software PHP”, (en línea; Junio de 2020). Disponible en:

https://www.php.net/license/3_01.txt

[22] BOE, “Reglamento General de Protección de Datos”, (en línea; Junio de 2020). Disponible en:

https://www.boe.es/doue/2016/119/L00001-00088.pdf

[23] Wikipedia, “Definición del Reglamento General de Protección de Datos”, (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Reglamento_General_de_Protecci%C3%B3n_de_Datos

[24] Wikipedia, “Definición de la distribución de Linux, CentOS”, (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/CentOS

[25] Wikipedia, “Definición de la herramienta de gestión de cortafuegos FirewallD”, (en línea; Junio de 2020). Disponible en:

https://en.wikipedia.org/wiki/Firewalld

[26] Wikipedia, “Definición del servidor XRDP”, (en línea; Junio de 2020). Disponible en:

https://en.wikipedia.org/wiki/Xrdp

[27] Wikipedia, “Definición del protocolo RDP”, (en línea; Junio de 2020). Disponible en:

https://en.wikipedia.org/wiki/Remote_Desktop_Protocol

[28] Wikipedia, “Definición del módulo de virtualización KVM”, (en línea; Junio de 2020). Disponible en:

https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine

[29] Wikipedia, “Definición del servidor web NGINX”, (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Nginx

[30] Wikipedia, “Características básicas del servidor web NGINX”, (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Nginx#Caracter%C3%ADsticas_b%C3%A1sicas_del_servidor_web

[31] Wikipedia, “Definición de la distribución XAMPP”, (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/XAMPP

[32] Wikipedia, “Definición del sistema de monitorización Zabbix”, (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Zabbix

[33] Enrique Rubio Royo y Antonio Ocón Carreras, “Enfoque de la red integral de la Universidad de Las Palmas de Gran Canaria”, (en línea; Junio de 2020). Disponible en:

https://www.rediris.es/difusion/publicaciones/boletin/31/enfoque2.html

[34] Wikipedia, “Definición del programa de transferencia de ficheros LFTP”, (en línea; Junio de 2020). Disponible en:

https://en.wikipedia.org/wiki/Lftp

[35] Wikipedia, “Definición del sistema de control de versiones Git”, (en línea; Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Git

[36] Riello, “Definición del software de gestión de UPS de Riello, PowerShield^{3}”, (en línea; Junio de 2020). Disponible en:

https://www.riello-ups.es/downloads/1-powershield

[37] ZLIB, “Definición de la herramienta de compresión de ficheros PIGZ”, (en línea; Junio de 2020). Disponible en:

https://zlib.net/pigz/

[38] Wikipedia, “Ventajas de la virtualización”, (en línea; de Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Virtualizaci%C3%B3n#Ventajas

[39] Wikipedia, “Ventajas de un proxy inverso”, (en línea; de Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Proxy_inverso

[40] DigitalOcean, “Guía de instalación de Zabbix en CentOS 7”, (en línea; de Junio de 2020). Disponible en:

https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-zabbix-to-securely-monitor-remote-servers-on-centos-7

[41] Jens Depuydt, “Configuración de dos interfaces de red en diferentes subredes en CentOS 7”, (en línea; de Junio de 2020). Disponible en:

http://jensd.be/468/linux/two-network-cards-rp_filter

[42] ItzGeek, “Guía de instalación de XRDP en CentOS 7”, (en línea; de Junio de 2020). Disponible en:

https://www.itzgeek.com/how-tos/linux/centos-how-tos/install-xrdp-on-centos-7-rhel-7.html

[43] NGINX, “Guía de administrador de NGINX como *proxy* inverso”, (en línea; de Junio de 2020). Disponible en:

https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/

[44] Moodle, “Guía de administración de Git”, (en línea; de Junio de 2020). Disponible en:

https://docs.moodle.org/all/es/Git_para_Administradores

[45] Wikipedia, “Definición de *Pre-Shared Key*”, (en línea; de Junio de 2020). Disponible en:

https://es.wikipedia.org/wiki/Pre-shared_key

[46] Zabbix, “Documentación de Zabbix”, (en línea; de Junio de 2020). Disponible en:

https://www.zabbix.com/manuals