



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE
SISTEMAS INFORMÁTICOS
UNIVERSIDAD POLITÉCNICA DE MADRID

GRADO EN INGENIERÍA DE COMPUTADORES
(PLAN 2014) - CURSO 2018/19

PROYECTO DE MIGRACIÓN DE CENTRO DE TRABAJO
BASADO EN WINDOWS A TECNOLOGÍAS LINUX

Autor: Eduardo Blázquez Málaga

Tutor: Andrés Sevilla de Pablo

7 de diciembre de 2018

Resumen

Actualmente multitud de centros de trabajo universitarios se encuentran implementados en base a tecnologías de código cerrado basadas en Windows. Utilizar este tipo de tecnologías implica ciertas desventajas, siendo la principal los costes de licencia asociado a dicho software. Por otro lado, Linux constituye una alternativa de sistema operativo libre bastante robusta, con un gran ecosistema de software disponible.

En este trabajo se propondrá un entorno de trabajo basado en equipos Linux, concretamente en la distribución CentOS. Se presentarán dos herramientas de gestión dedicadas para entornos Linux: FreeIPA para la gestión de usuarios como alternativa a Microsoft Active Directory, y Foreman para el despliegue automatizado y la gestión masiva de los equipos de trabajo. Además se proporcionará una arquitectura sencilla para esta solución, y se describirán los pasos llevados a cabo para su instalación en un entorno de pruebas, en el que se comprobará el correcto funcionamiento y la viabilidad de la misma.

Se proporcionarán además soluciones para la ejecución de software típico empleado en un centro de trabajo universitario, o alternativas de uso para los casos en que no sea viable utilizar un software en concreto. Posteriormente, se propondrá un plan de trabajo de unas 18 semanas para llevar a cabo una implantación de un entorno de estas características. Se sugerirá también el uso del software de gestión para otros entornos del centro universitario que puedan utilizar Linux, como su centro de datos o ciertos laboratorios de trabajo.

Finalmente se discutirá acerca de los beneficios de una hipotética migración de un ecosistema Windows a Linux, y se presentarán además casos reales de ahorros en licencias, que pueden llegar incluso a millones de euros en grandes entornos.

Abstract

Nowadays, multiple computer centres in universities are implemented using closed-source technologies based on Windows. The utilisation of these kind of technologies implies several disadvantages, being the main one the license costs associated to this software. On the other hand, the Linux operating system is a pretty robust alternative, and possesses a large collection of free software.

In this paper, a work environment based on Linux will be proposed, specifically using the CentOS distribution. Two management tools for Linux environments will be presented: FreeIPA for user management (as an alternative to Microsoft Active Directory), and Foreman for automated deployments and management of desktop systems. Furthermore, a simple architecture for such a solution will be described, as well as the steps taken for its installation on a test environment, where its proper function and viability will be verified.

Solutions will be provided to run typical software used in a university computer centre, or execution alternatives for specific programs that would not work on Linux. Afterwards, a work schedule of about 18 weeks will be proposed to carry out the implementation of this kind of environment. Also, additional use cases will be suggested for the management software, specifically for other areas of the university that may use Linux, such as computing laboratories or its data centre.

Finally, the benefits of a hypothetical migration from Windows to Linux will be discussed, including real-life license savings scenarios, which could reach millions of euros on large environments.

Índice

1. Introducción	5
2. Objetivos y beneficios	7
3. Especificaciones de la solución	9
3.1. Sistema operativo	9
3.2. Sistema de gestión de usuarios	10
3.3. Sistema de aprovisionamiento	11
3.4. Consideraciones de almacenamiento	13
3.5. Arquitectura de la solución	13
4. Instalación de la solución	16
4.1. Entorno de instalación	16
4.2. Instalación de FreeIPA	17
4.3. Instalación de Foreman	26
4.4. Instalación de puesto de trabajo	32
4.5. Diagrama de secuencia de aprovisionamiento	37
5. Compatibilidad y alternativas de software	38
5.1. Software de propósito general	38
5.2. Entornos de desarrollo	39
5.3. Entornos de ejecución	40
5.4. Programación de hardware	41
5.5. Otras utilidades de diseño y desarrollo	41
5.6. Herramientas de redes	42
5.7. Herramientas de seguridad	43
5.8. Bases de datos y almacenamiento	43
5.9. Software de virtualización	44
5.10. Software matemático	45
5.11. Directrices generales de ejecución de software sobre Linux	45
5.12. Tabla resumen de software	46
6. Estimación de tiempos	49
7. Otros usos	50
8. Impactos sociales y ambientales	51
9. Responsabilidad ética y profesional	52
10. Conclusiones	53
11. Bibliografía	54

12. Anexos	56
12.1. Anexo 1 - Acrónimos	56
12.2. Anexo 2 - URLs con compatibilidad de software	58

Índice de figuras

1.	Ejemplo de arquitectura simple	15
2.	Pantalla de creación de máquina virtual	17
3.	Opciones de instalación de CentOS	18
4.	Extracto de ejecución de instalador de FreeIPA	19
5.	Pantalla inicial de FreeIPA	19
6.	Creación de grupo de alumnos	20
7.	Creación de usuario <i>alumno</i>	21
8.	Regla de HBAC para grupo de alumnos	22
9.	Extracto de regla de <i>sudo</i> para grupo de administradores	23
10.	Prueba de acceso con usuarios de FreeIPA	23
11.	Creación de <i>map</i> para directorios de usuario	24
12.	Creación de clave para directorios de usuario	25
13.	Acceso con usuario tras habilitar <i>automount</i>	25
14.	Regla de automembresía para grupo de hosts <i>puestos</i>	26
15.	Extracto de registro de Foreman en FreeIPA	27
16.	Opciones de instalación de Foreman	27
17.	Pantalla inicial de Foreman	28
18.	Datos de creación de sistema operativo en Foreman	29
19.	Creación de subred en Foreman	30
20.	Extracto de script de integración de Foreman y FreeIPA	31
21.	Creación de reino de Kerberos en Foreman	31
22.	Configuración de sistema operativo para grupo de hosts <i>puestos</i>	32
23.	Dirección MAC de la tarjeta de red	33
24.	Configuración de interfaz de red de Foreman	34
25.	Arranque por red mediante PXE	35
26.	Comprobación de correcto funcionamiento de VM aprovisionada	35
27.	Diagrama de secuencia de aprovisionamiento	37
28.	Diagrama de Gantt con estimación de tiempos	49

Capítulo 1. Introducción

En la actualidad gran parte de las infraestructuras informáticas del mundo están soportadas por Linux y tecnologías libres y de código abierto. Las ventajas de usar software de este estilo son varias. Algunos ejemplos serían los siguientes: gran eficiencia, costes muy bajos (o incluso nulos), y una gran comunidad de desarrolladores y colaboradores.

En los centros de datos empresariales encontramos Linux desde hace ya bastantes años, incluyendo entornos muy críticos (como por ejemplo en la industria bancaria), lo que puede dar una idea de la fiabilidad de dicho sistema operativo. También hemos visto como en la lista de los ordenadores más potentes del mundo, el TOP 500, Linux ha ido reemplazando poco a poco a Unix como sistema operativo favorito, hasta llegar a un 100 % de utilización en noviembre de 2017 [1].

Aunque podamos pensar que Linux no tiene gran acogida para el usuario de a pie (principalmente por su falta de uso en equipos de escritorio), encontramos en el mercado de dispositivos móviles una gran predominancia de dispositivos Android, sistema operativo que corre bajo un núcleo Linux modificado. Esto puede dar una idea de la versatilidad de este sistema operativo, y desterrar la imagen de dificultad en su manejo a la que generalmente se le asocia.

El desarrollo de Linux y su núcleo no solo lo llevan a cabo desarrolladores y contribuyentes anónimos; grandes empresas informáticas también colaboran en dicho desarrollo. Éste es el caso de, entre otros, Intel y Red Hat, que según un informe de la Linux Foundation (fundación para promover el uso de Linux) realizaron conjuntamente alrededor de un 21 % de cambios en el código del kernel entre 2015 y 2016 [2]. Con esto podemos observar que hay grandes empresas trabajando de forma continuada para hacer que Linux sea cada vez más eficiente, estable y seguro.

Es cierto que en otros ámbitos Linux no posee un elevado nivel de implantación, por ejemplo en el caso de los entornos de escritorio que indicábamos. Consideramos que hay dos factores principales que provocan que los usuarios prefieran un sistema operativo Windows antes que Linux: en primer lugar la ausencia de ciertos programas bien conocidos por los usuarios (suites de productividad, programas de edición de imágenes...), y en segundo la experiencia ya existente del usuario con el entorno Windows, combinado con la imagen de dificultad que pueda dar Linux

La idea de falta de software en Linux es por supuesto una falacia, ya que este sistema operativo está rodeado por un amplio ecosistema de paquetes y programas de múltiples categorías, que además se encuentran disponibles en repositorios fácilmente accesibles para su instalación.

Si bien es cierto que hay aplicaciones concretas que no podemos encontrar en Linux, tendremos casi siempre una alternativa de código abierto (más o menos válida, según el caso) a estos programas con nombre y apellidos; además podremos utilizar otros mecanismos para ejecutar nuestro software predilecto, como emuladores o máquinas virtuales. De hecho en este mismo documento se estudiarán ejemplos de programas que se ejecutan bajo Windows, y diferentes alternativas de software o de ejecución.

También es cierto que los usuarios que tengan experiencia en entorno Windows van a ser más reacios a cambiar de sistema operativo, ya que esto implica un proceso de aprendizaje. Curiosamente, este caso se produce con Linux, pero no con ordenadores Apple (los cuales usan macOS, sistema operativo derivado de Unix). Mientras que Apple proyecta una imagen de sencillez, Linux parece justo lo contrario, lo que puede provocar el rechazo de los usuarios.

Esto puede que fuera cierto hace años, ya que incluso el proceso de instalación de ciertas distribuciones (Debian, por ejemplo) podía resultar complejo y confuso, incluso para gente con conocimientos informáticos. Sin embargo esto ya no es así: ahora mismo podemos encontrar muchas distribuciones de Linux cuyo manejo está precisamente diseñado para que resulte sencillo y amigable al usuario (por ejemplo Ubuntu o Linux Mint).

Por otro lado, en entornos educativos todavía encontramos un gran nivel de implantación de tecnologías Windows. Es posible que para ciertas materias resulte más cómodo utilizar Windows (principalmente por necesidad de uso de ciertos programas), pero para la enseñanza en materia de informática el uso de dicho sistema operativo conlleva ciertos inconvenientes, por ejemplo por el mero hecho que el uso de Windows en sí mismo generalmente no resulta didáctico en absoluto.

Otro problema derivado del uso de Windows sería su coste de uso, ya que al tratarse de software propietario estamos obligados a pagar una serie de licencias por cada puesto de trabajo de que dispongamos. Es posible que en centros educativos privados esto resulte aceptable, pero cuando hablamos de centros públicos podemos encontrarnos con problemas de presupuesto. Adicionalmente, al tratarse de código cerrado, hay funcionalidades del software que solo el fabricante conoce, lo que puede facilitar la implementación de puertas traseras o *backdoors*, por motivos que pueden ir desde el control del propio software hasta espionaje empresarial o gubernamental [3].

Por estos motivos, lo más adecuado para un entorno educativo sería utilizar tecnologías libres y de código abierto. El mayor problema para llevar a cabo este cambio radica en la dificultad de una migración completa de un centro de trabajo a dichas tecnologías, lo que por otro lado continúa ayudando a la perpetuación de Windows como sistema operativo predominante.

Deberemos por tanto encontrar una serie de tecnologías y metodologías que permitan una migración gradual y sencilla de equipos existentes, de forma que el proceso de migración tenga el menor impacto posible, y ofrezca las mismas funcionalidades de las que se pueda disponer actualmente en un ecosistema Windows. Éste será el propósito de este trabajo.

Capítulo 2. Objetivos y beneficios

Como avanzábamos en la introducción, el objetivo de este trabajo es el de proponer una serie de mecanismos para lograr de manera exitosa una migración de un centro de trabajo (principalmente enfocado al ámbito universitario) de un entorno existente con tecnologías Windows, a otro basado en tecnologías open-source corriendo bajo un sistema operativo Linux.

A lo largo de este trabajo iremos tratando diversas cuestiones relacionadas con la migración. En primer lugar introduciremos ciertos detalles sobre el sistema operativo Linux con el objetivo de presentar sus principales características y ventajas e inconvenientes en comparación con Windows. Discutiremos también acerca de las diferentes familias y distribuciones de Linux, y presentaremos una propuesta de una distribución concreta argumentando dicha elección.

Posteriormente presentaremos una serie de herramientas adecuadas para la gestión centralizada de una infraestructura Linux, de manera que el control de los diferentes equipos se pueda realizar de una forma sencilla, y de forma que puedan suplirse las herramientas actuales que se puedan estar empleando en el centro de trabajo.

A este respecto presentaremos inicialmente un sistema de gestión de usuarios para Linux, de forma que podamos suplir la funcionalidad del conocido Active Directory de Microsoft. Dicho sistema de gestión deberá integrarse con Active Directory, de forma que ambas soluciones convivan en un primer momento y que haya consistencia entre los usuarios de ambas plataformas, para finalmente (cuando se pueda prescindir del AD) migrar todos los usuarios existentes a este sistema.

A continuación presentaremos una herramienta para ayudar con el despliegue de Linux sobre los diferentes equipos, de forma que el proceso de migración sea realmente simple, y que la instalación de Linux se realice de forma automática para facilitar el proceso de migración. Igualmente mediante esta herramienta podremos gestionar instalaciones de software y actualizaciones, todo ello de forma centralizada.

Tras esto discutiremos ciertas consideraciones sobre el almacenamiento compartido, utilizado para que los usuarios puedan disponer de sus datos entre los equipos del centro. No entraremos demasiado en profundidad en esta materia, ya que generalmente las soluciones utilizadas para este almacenamiento son agnósticas con respecto a la tecnología utilizada, y proporcionan la capacidad de compartir unidades tanto para entornos Windows como para entornos Linux.

Por último, en lo que respecta al proceso de migración, discutiremos acerca del software que pueda utilizarse en el entorno universitario. Hablaremos de software existente y las

diferentes posibilidades que puedan existir para su manejo en un entorno Linux. Cabrá la posibilidad que ciertos programas puedan manejarse de forma nativa en Linux, pero habrá ciertos casos en los que no; en dichos casos recomendaremos software alternativo, u otras vías de ejecución para que su uso sea factible.

Los beneficios que tratamos de obtener con una infraestructura como la que se describe en este PFG serán múltiples: como ya hemos avanzado un entorno basado en tecnologías de código abierto será por definición más económico que un entorno basado en tecnologías propietarias. Se pretende también el poder disponer de un sistema de gestión centralizado, de forma que múltiples equipos puedan ser instalados, configurados, actualizados o reemplazados con un mínimo esfuerzo de gestión. Adicionalmente, se desea que los alumnos puedan disfrutar de un entorno más eficiente sobre el que trabajar, y más atractivo a nivel didáctico.

La definición del entorno propuesto será semejante al empleado en entornos empresariales para la gestión masiva de servidores Linux. Se tendrán consideraciones adicionales para este PFG, especialmente por el hecho que éste estará centrado en equipos de escritorio en lugar de servidores, y que el entorno es un centro universitario, no empresarial.

Capítulo 3. Especificaciones de la solución

En esta sección describiremos la solución propuesta para el despliegue y gestión de una infraestructura basada en equipos Linux. Se proporcionarán detalles de cada componente de la solución, comenzando por el sistema operativo que utilizaremos en los puestos de trabajo (CentOS), y siguiendo con los sistemas de gestión de usuarios (FreeIPA) y equipos (Foreman).

Se proporcionarán también ciertos detalles sobre el almacenamiento de cara a disponer de los datos de cada usuario en todos los equipos, y finalmente se darán una serie de consideraciones acerca de la arquitectura de la solución, proponiendo además un modelo sencillo de arquitectura.

Apartado 3.1. Sistema operativo

En primer lugar vamos a hablar sobre el sistema operativo elegido, o mejor dicho, la distribución de Linux concreta que se ha elegido tanto para implementar la infraestructura de servidores, como para los equipos de escritorio de alumnos que se desplegarán mediante dicha infraestructura. Esta distribución será CentOS [4].

Las distribuciones de Linux se separan en dos grandes familias, las basadas en Red Hat, y las basadas en Debian. Aunque ambas familias tengan bastantes diferencias, la principal de todas ellas es el formato de paquetes utilizado: *rpm* en el caso de Red Hat, y *dpkg* en el de Debian. La mayoría de distribuciones derivan de estas dos; por ejemplo la conocida distribución Ubuntu está basada en Debian. Comentar que hay muchas otras familias, pero generalmente no tienen tanta relevancia y no suelen utilizarse en entornos productivos.

CentOS (Community Enterprise Operating System) se trata de una distribución de Linux basada en Red Hat Enterprise Linux (o RHEL), desarrollada por Red Hat [5], que es actualmente una de las distribuciones más importantes a nivel empresarial. Por consiguiente, CentOS posee una gran estabilidad en cuanto a versiones de software incluido en la distribución, aunque por otro lado gran velocidad en el lanzamiento de parches críticos de seguridad o de corrección de errores.

La numeración de versiones de la distribución coincide con la de RHEL, con una numeración tipo *major* y *minor*, separadas por un punto. La versión *major* identifica cambios bastante significativos en la distribución, mientras que la *minor* identifica pequeños cambios, como ciertas actualizaciones para mejorar compatibilidades o incluir nuevas características. En el momento de escribir este PFG la versión más reciente es la 7.5.

Acerca de las diferencias, la más notable es que CentOS es un sistema operativo totalmente gratuito, mientras que RHEL (a pesar de ser de código abierto), se trata de un sistema de pago, ya que para acceder a los repositorios o descargas del software ya

compilado requerimos una suscripción. Adicionalmente, las suscripciones de RHEL nos permitirán acceder a un soporte de nivel empresarial, mientras que para CentOS solamente contaremos con el soporte de la comunidad, o bien esperar a que los cambios realizados en RHEL se transmitan a CentOS.

La elección de un sistema operativo de familia Red Hat en lugar de Debian ha sido tomada en base al gran ecosistema de software de gestión que existe para ésta, ya que Red Hat colabora en el desarrollo de dicho software para potenciar el uso de su sistema operativo en entornos empresariales. Entre dicho software se encuentra el que se propone en este documento, como Foreman para la gestión centralizada de máquinas, o FreeIPA para la gestión de usuarios.

Además se ha elegido CentOS puesto que dentro de la familia Red Hat es la distribución que posee más movimiento a nivel comunitario. Inicialmente se planteó el uso de Scientific Linux, otra distribución similar desarrollada precisamente para entornos universitarios y de investigación. Sin embargo, parece que esta distribución cada vez recibe menos mantenimiento y se tarda bastante en portar ciertos parches. Además el hecho que el CERN, uno de los desarrolladores de la distribución, decidiera en 2015 reemplazar Scientific Linux por CentOS [6] facilitó la decisión del uso de este último.

Existe otra distribución de la familia Red Hat muy popular, llamada Fedora [7]. Sin embargo dicha distribución es menos estable, ya que en ella se suelen introducir frecuentemente cambios y nuevas funcionalidades, puesto que se utiliza principalmente como rama de pruebas de la distribución Red Hat. Es por ello que tampoco hemos propuesto su uso para el entorno que queremos implementar.

Apartado 3.2. Sistema de gestión de usuarios

Dado que la idea del proyecto es migrar una infraestructura Windows, y que una de las piezas claves de dicha infraestructura suele ser el Directorio Activo (también conocido como Active Directory, o AD), el nuevo modelo propuesto para un entorno Linux debe proveer de una solución similar para la gestión centralizada de usuarios.

La solución propuesta en este caso es FreeIPA [8], un software libre y gratuito para gestión de identidades, también respaldado por Red Hat. Se trata de una solución que combina ciertos protocolos de autenticación, como LDAP o Kerberos (usados también por Active Directory), para identificar y autorizar usuarios de forma segura en toda la plataforma.

FreeIPA no solo permite gestionar de forma remota usuarios, credenciales, políticas de contraseña, sino que además permite realizar un control de acceso más granular, limitando el acceso de ciertos usuarios a un conjunto de máquinas (metodología conocida como *Host-based Access Control* o HBAC). Es posible además crear grupos tanto de usuarios como de máquinas, y de esta forma definir dichas políticas en base a grupos.

Si bien los sistemas Linux pueden conectarse directamente contra un AD mediante LDAP y Kerberos para gestionar la autenticación de usuarios, hay ciertos permisos del usuario que hay que gestionar individualmente en cada máquina, como los permisos de superusuario, proporcionados mediante el programa *sudo*. La ventaja de FreeIPA a este respecto es que también permite centralizar dichos permisos, de forma que no tengamos que desplegar una configuración para *sudo* en cada una de las máquinas.

FreeIPA posee además un servidor DNS integrado con el que puede gestionar las resoluciones de nombres para la infraestructura de equipos de escritorio. Sería posible crear un subdominio exclusivo para éste, y configurar los servidores DNS principales para que deleguen las consultas de dicho subdominio a los servidores de FreeIPA.

Evidentemente resulta complicado pensar que FreeIPA pueda reemplazar de forma inmediata a un Directorio Activo, por ejemplo durante el proceso de migración, o que resulte directamente imposible en caso que se siga manteniendo parte de la infraestructura sobre Windows. Esto llevaría a un escenario en el que tendríamos dos sistemas diferentes de gestión de usuarios, que habría que mantener por separado, lo que dificultaría la gestión de la plataforma.

Afortunadamente FreeIPA permite dos formas de comunicación con AD, o bien sincronizando los usuarios que tiene registrado este último, o bien establecer una relación de confianza entre ambos, de forma que FreeIPA transfiera las peticiones de autenticación al AD. Esto permitiría una transición bastante sencilla entre ambas tecnologías.

Finalmente, si fuera posible migrar todos los sistemas Windows, podría eliminarse la infraestructura del Directorio Activo y mantener únicamente FreeIPA. Comentar por último que, en caso que el AD también esté proporcionando servicios de DNS, habría que reemplazarlo por un nuevo DNS para el dominio principal, e igualmente configurarlo para que delegue las consultas del subdominio de FreeIPA.

Apartado 3.3. Sistema de aprovisionamiento

Para poder materializar un proyecto de estas características, sería de gran ayuda disponer de un mecanismo de instalación y configuración automatizada de los puestos de trabajo. El mero hecho de tener que realizar las tareas de instalación y configuración inicial de forma manual en cada equipo puede resultar un impedimento para la migración, además que se trata de una serie de tareas repetitivas, bastante propensas a errores.

Foreman [9] es un proyecto, también apoyado por Red Hat, que permite aprovisionar equipos de forma automática. Esto facilitaría enormemente el procedimiento de migración, y serviría también para realizar nuevas instalaciones en caso que sea necesario ampliar el número de equipos o reemplazar alguno de ellos. Foreman nos permite crear imágenes de disco con una configuración de instalación ya definida, o aprovisionar equipos por red mediante protocolo PXE. De esta forma, los equipos en el arranque se comunicarían

con el servidor de Foreman, y solicitarían una imagen y una configuración de instalación (particionado del equipo, listado de paquetes a instalar, etcétera).

Para aplicar esta configuración, el instalador de CentOS (Anaconda) permite utilizar un formato de fichero de respuestas (conocido como Kickstart), en el que especificaremos la configuración inicial de la máquina. Dentro de Foreman dispondremos de diferentes plantillas de Kickstart que podremos personalizar para nuestra instalación. Dichas plantillas se servirán luego por HTTP para que el instalador pueda recuperarlo a través de la red.

Foreman dispone además de multitud de *plugins* para ampliar sus capacidades. Por ejemplo mediante Pulp y Katello es posible sincronizar los repositorios públicos de CentOS en el servidor de Foreman, de forma que los puestos de trabajo no necesiten acceder al exterior para recuperar paquetería del sistema. Esto puede resultar especialmente útil en entornos en los que no se permite salida directa a Internet desde los puestos de trabajo.

Además Katello permite gestionar el ciclo de vida de los repositorios, de forma que es posible mantener un conjunto de versiones de paquetería de forma estática para asegurarse que las versiones de software de los equipos son consistentes entre sí. También permite promocionar versiones a diferentes entornos, lo que permite probar un conjunto de versiones en un entorno de preproducción, y posteriormente promocionarlo a un entorno de producción, aunque bien es cierto que esta funcionalidad es más útil en entornos de servidores que de puestos de trabajo.

Podemos utilizar además los *plugins* de gestión de repositorios para crear repositorios propios, por ejemplo para instalar software utilizado en el centro que no se encontrara disponible mediante repositorios ni estuviera empaquetado en RPM. Eso sí, para ello habría que generar un paquete RPM por cada programa, lo que puede resultar tedioso. El empaquetado en RPM se encuentra fuera del alcance de este papel.

También es posible instalar otros *plugins* útiles, como por ejemplo para realizar gestión de configuración con software como Ansible o Puppet. De esta forma podríamos aplicar una configuración más compleja tras la instalación, centralizar toda la configuración de las máquinas y ser capaces de aplicar nuevas configuraciones en cualquier momento, y poder comprobar periódicamente que la configuración de los puestos de trabajo es la correcta.

En principio se barajó la posibilidad de utilizar Spacewalk [10] como sistema de aprovisionamiento y gestión de ciclo de vida de los equipos de trabajo. Sin embargo las capacidades de aprovisionamiento de Spacewalk son muy limitadas, ya que está más orientado a la gestión del ciclo de vida del software de las máquinas registradas. Además, el sistema de gestión empresarial de Red Hat (Red Hat Satellite) pasó en su versión 5 de estar basado en Spacewalk, a estarlo en Foreman en su versión 6. Teniendo en cuenta que Red Hat mantiene ambos proyectos, es de suponer que se dedicará menos esfuerzo de mantenimiento y soporte a Spacewalk que a Foreman, que es la apuesta actual de Red Hat.

Apartado 3.4. Consideraciones de almacenamiento

Para que cada usuario pueda conectar a diferentes puestos de trabajo, es necesario que parte del almacenamiento utilizado por dichos puestos resida en un servidor de almacenamiento externo, en el que cada usuario almacenará sus documentos personales o preferencias de aplicación.

En este papel no se pretende reemplazar la solución de almacenamiento empleada, ya que típicamente se utilizará un sistema NAS dedicado. Éstos pueden servir almacenamiento por diferentes protocolos, como SMB (usado principalmente en sistemas Windows) o NFS (usado en sistemas Linux/UNIX), con lo que el mismo sistema nos servirá para ambos ecosistemas.

Aunque los sistemas Linux pueden emplear unidades de red SMB mediante Samba (una implementación libre de dicho protocolo), lo más recomendado es que los ficheros sean servidos por NFS. Esto es debido a que NFS es compatible con el estándar de permisos de Linux, lo que permite un control de acceso a los diferentes ficheros más granular y más seguro.

Aprovechamos también para indicar otra ventaja que obtendríamos con Linux al respecto del almacenamiento, y es que mientras que típicamente en un entorno Windows la compartición de los datos de usuario es compleja, y generalmente se limita a una carpeta compartida donde almacenar ficheros, en Linux es mucho más sencilla al tener que compartir únicamente el directorio *HOME* del usuario. En este directorio se almacenan no solo ficheros de usuario, sino también preferencias de aplicación, escritorio, etc., con lo cual la experiencia del usuario en un equipo u otro es prácticamente la misma.

Apartado 3.5. Arquitectura de la solución

De cara a definir la arquitectura de la solución, hay una serie de consideraciones que debemos tener en cuenta:

- Los equipos de escritorio se encontrarán en un segmento de red diferente al del resto de la infraestructura, ya que aprovisionaremos los mismos mediante DHCP/PXE y no queremos que el mecanismo de provisión pueda afectar a otros equipos.
- El servidor de Foreman será el que proporcione dicha provisión, así que deberá encontrarse en el segmento de red indicado en el punto anterior. Foreman gestionará el arranque por PXE y típicamente el DHCP, aunque sería viable utilizar un servidor DHCP externo (como un router).
- El segmento de red de provisión deberá tener salida al exterior para la descarga de paquetería del sistema operativo durante la instalación de los equipos. La conectividad al exterior se podrá realizar o bien enrutando hacia la red pública, o bien mediante un proxy con salida al exterior.

- Como alternativa, sería posible utilizar el servidor de Foreman como repositorio para la paquetería, aunque esto tendría sentido principalmente si los equipos no pudieran tener salida a Internet. Dado que por lo general sí tendrán salida al exterior, utilizaremos los repositorios públicos de CentOS. Más adelante podrían servirse mediante Foreman para reducir el tráfico al exterior.
- Dado que el servidor de Foreman se utilizará para tareas puntuales, como desplegar equipos, o instalar/actualizar paquetería en los sistemas gestionados, no se considerará un servicio crítico. Es por ello que no será necesario proporcionar alta disponibilidad para el mismo (aunque sí deberemos realizar backups periódicos en caso que sea necesario recuperar el servidor).
- En la red de gestión se encontrarán las máquinas que vayan a proporcionar servicios internos a la infraestructura de puestos de trabajo, como servidores de nombres, almacenamiento o gestión de usuarios. Típicamente usaremos la red de gestión para administrar estos servicios, así que el servidor de Foreman podrá tener también conectividad en esta red para su administración.
- Dado que el servicio de FreeIPA es necesario para la autenticación de usuarios, será necesario disponer de al menos dos servidores de FreeIPA para garantizar la alta disponibilidad del servicio.
- FreeIPA deberá encontrarse en un dominio DNS que no entre en conflicto con el del AD, ya que para establecer una relación a nivel de Kerberos entre ambos deberán encontrarse en diferentes dominios.
- El enrutamiento entre las diferentes redes debería pasar por un cortafuegos para permitir únicamente el acceso de los puestos de trabajo a los servicios necesarios, y evitar accesos indeseados (como acceso por SSH de un puesto de trabajo al servidor de autenticación).

A continuación mostramos un ejemplo sencillo de arquitectura:

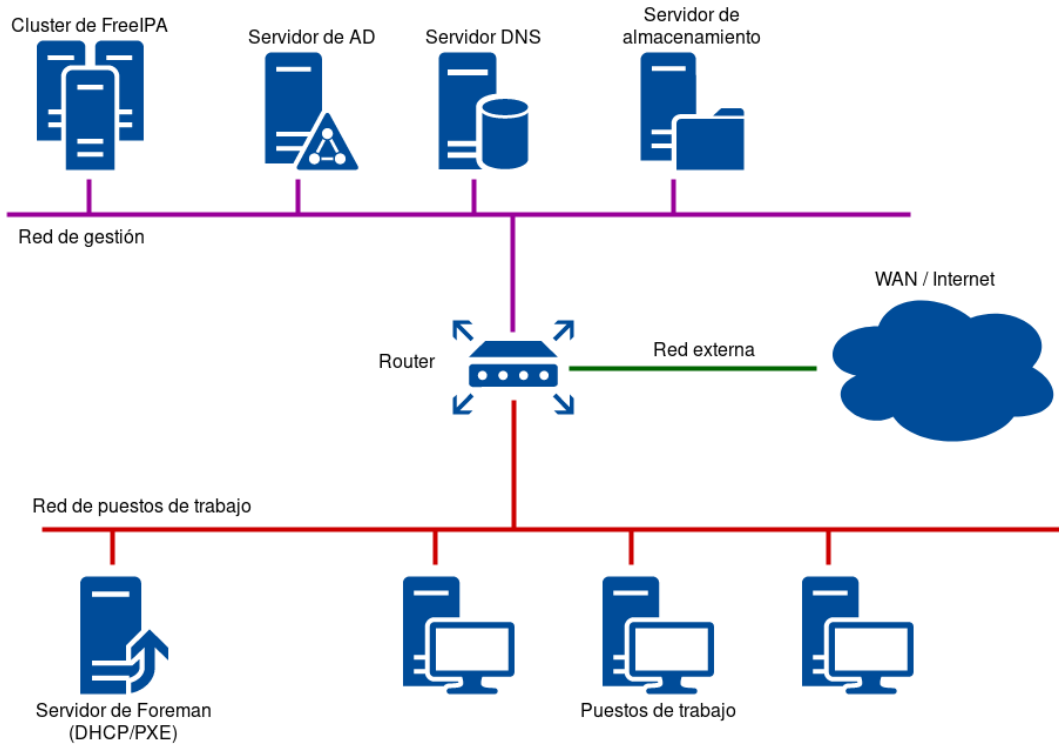


Figura 1: Ejemplo de arquitectura simple

Entendemos que el Active Directory se encuentra también en cluster, pero lo mostramos como un servidor por simplicidad, dado que la arquitectura del AD se encuentra fuera del alcance de este documento. Igualmente, FreeIPA aparece como un cluster de tres servidores, aunque podrían emplearse dos.

A nivel de red aparece un único router, pero entendemos que entre las diferentes redes podrán existir diferentes routers, firewalls y demás elementos de electrónica de red. Por último, indicar que el servidor de Foreman se ha ubicado en la parte inferior por mostrar un esquema sencillo, pero también debería tener conectividad directa en la red de gestión.

Para más consideraciones de arquitectura acerca de FreeIPA, recomendamos consultar la guía oficial de recomendaciones de despliegue [11].

Capítulo 4. Instalación de la solución

En esta sección se describirá la instalación realizada en un entorno de pruebas, en el que desplegaremos los diferentes componentes de la arquitectura propuesta. Inicialmente, se llevará a cabo la instalación de los servidores de Foreman y FreeIPA (ambos en máquinas virtuales), y posteriormente probaremos a desplegar nuevos escritorios de forma automática. Para el almacenamiento de los directorios de usuario utilizaremos por simplicidad el servidor de virtualización directamente en lugar de un servidor dedicado.

Apartado 4.1. Entorno de instalación

Las máquinas virtuales de pruebas las instalaremos en un servidor de las siguientes características:

- CPU: Intel Core i7-3770 (4 cores, 8 threads, 3.40GHz)
- RAM: 16GB (4 x 4GB) Kingston HyperX DDR3-1600 CL9 SDRAM
- Disco: Seagate Barracuda 7200.14 1TB SATA3
- Red: Adaptador wifi Edimax EW-7811Un IEEE802.11b/g/n

En este servidor disponemos de una instalación de Arch Linux, una distribución de Linux muy ligera y altamente configurable. La virtualización de equipos la realizaremos mediante KVM, un módulo de kernel que convierte al equipo en un hipervisor tipo 1 [12], es decir, que tendremos una virtualización más cercana al hardware y que hace uso de las extensiones de virtualización del procesador.

El servidor se encuentra en una red local con un rango 192.168.1.0/24. Además se creará una red virtual con rango 192.168.100.0/24, en el que el hipervisor tendrá la dirección IP 192.168.100.1. El hipervisor ejercerá de *gateway* para las VMs, y realizará un NAT para que éstas puedan comunicarse con el exterior, ya que el rango 192.168.100.0/24 es un rango de red privado. Sin embargo el hipervisor no funcionará como servidor DHCP, ya que estas funciones se delegarán en el servidor de Foreman.

Este servidor también exportará mediante NFSv4 el directorio */userdirs*, que servirá como base para los *HOME* de los diferentes usuarios. Por tanto las VMs podrán acceder a este directorio a través de la unidad de red ubicada en 192.168.100.1:/userdirs. Por seguridad se ha limitado el montaje a los clientes que se encuentren en la red 192.168.100.0/24 (la virtual).

Apartado 4.2. Instalación de FreeIPA

En primer lugar procederemos a crear la máquina virtual para el servidor de FreeIPA. Utilizaremos una imagen de instalación mediante red de CentOS 7.5. Asignaremos a la VM 2 CPUs virtuales, 2GB de RAM y un disco de 10GB.

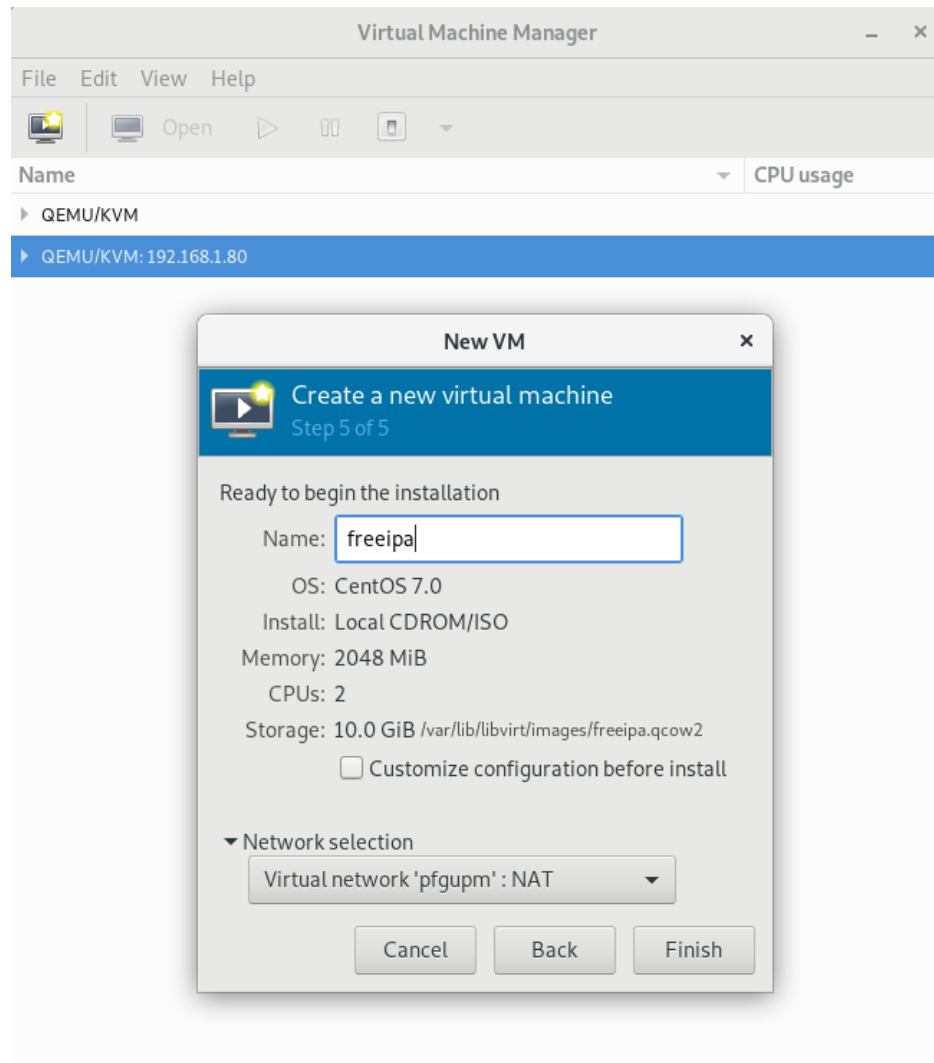


Figura 2: Pantalla de creación de máquina virtual

Esto nos cargará una imagen de CentOS, en la que seleccionaremos la opción de instalación. Tras elegir el idioma de instalación configuraremos diversos parámetros relevantes: zona horaria, distribución de teclado, repositorio principal, particionado y datos de red. Para la configuración de red hemos seleccionado la dirección IP 192.168.100.2, y el nombre de máquina *freeipa.pfgupm.local*. Una vez configurada la red, seleccionaremos como fuente de instalación la URL del repositorio público de CentOS (http://mirror.centos.org/centos-7/7/os/x86_64/).

Acerca de la selección de software elegiremos el tipo de instalación mínima, ya que instalaremos manualmente Foreman a través de la consola. Sobre el particionado hemos elegido particionado automático, ya que para nuestro entorno de pruebas no necesitamos ningún tipo de particionado avanzado.

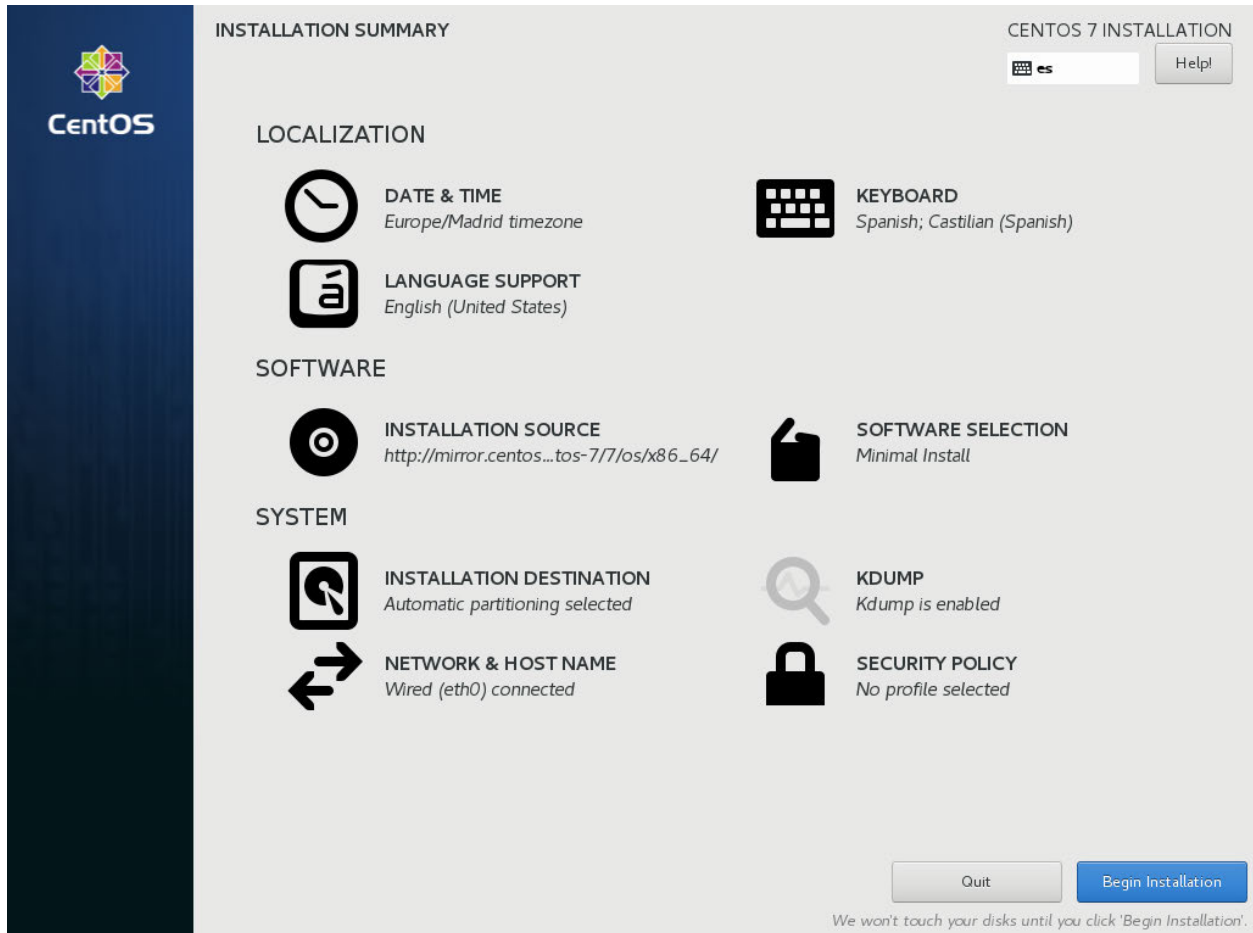


Figura 3: Opciones de instalación de CentOS

Durante la instalación se nos solicitará la contraseña del usuario administrador (*root*) y se nos proporcionará también la posibilidad de crear un nuevo usuario. De momento para esta instalación no crearemos ningún usuario adicional, aunque una buena práctica es la de crear un usuario administrador diferente, para poder deshabilitar el acceso remoto directo como *root*. Tras la instalación añadiremos en el fichero */etc/hosts* una entrada con la IP y los nombres completo (FQDN) y corto del servidor. Por último realizaremos una actualización completa de la VM mediante el comando `yum update -y` y reiniciaremos la misma.

Para la instalación de FreeIPA deberemos instalar los paquetes *ipa-server* e *ipa-server-dns*, y a continuación ejecutar el comando `ipa-server-install`, tal y como se especifica en la guía de instalación [13]:

```
[root@freeipa ~]# ipa-server-install --hostname=freeipa.pfgupm.local --domain pfgupm.local --realm=PFGUPM.LOCAL \
> --no-host-dns --setup-dns --auto-forwarders --auto-reverse
Checking DNS domain pfgupm.local, please wait ...

The log file for this installation can be found in /var/log/ipaserver-install.log
=====
This program will set up the IPA Server.

This includes:
* Configure a stand-alone CA (dogtag) for certificate management
* Configure the Network Time Daemon (ntpd)
* Create and configure an instance of Directory Server
* Create and configure a Kerberos Key Distribution Center (KDC)
* Configure Apache (httpd)
* Configure DNS (bind)
* Configure the KDC to enable PKINIT
```

Figura 4: Extracto de ejecución de instalador de FreeIPA

Dado que en CentOS 7 está habilitado por defecto el servicio de cortafuegos *firewalld*, deberemos incluir las reglas necesarias para permitir la conectividad hacia el servidor de FreeIPA. Para ello simplemente tendremos que permitir los servicios *freeipa-ldap*, *freeipa-ldaps* y *dns* usando el comando *firewall-cmd --add-service=<servicio>*. Podremos guardar esta configuración para posteriores reinicios con el comando *firewall-cmd --runtime-to-permanent*.

Tras la instalación podremos acceder mediante web al portal de administración de FreeIPA, que muestra el siguiente aspecto:

Figura 5: Pantalla inicial de FreeIPA

Podremos ver en el menú superior de la interfaz diferentes secciones. En cada una gestionaremos diferentes elementos:

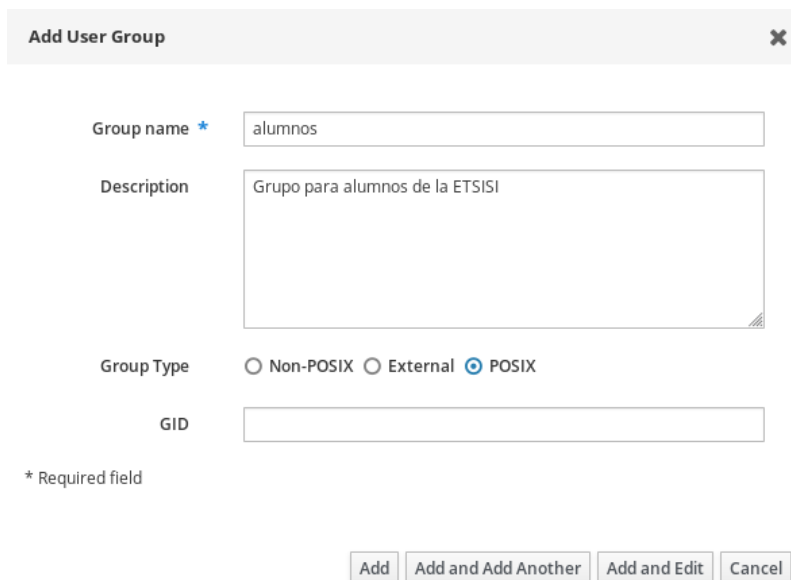
- *Identity*: usuarios y estaciones registradas en FreeIPA, y gestión de grupos de los anteriores.
- *Policy*: permisos de acceso y de *sudo*, y políticas de contraseña.

- *Authentication*: gestión de otros mecanismos de autenticación (certificados, tokens OTP, etc.).
- *Network Services*: gestión del DNS de FreeIPA y opciones para el servicio *Automount*.
- *IPA Server*: opciones de administración del servidor de FreeIPA.

Para el entorno que nos atañe utilizaremos una configuración sencilla, y emplearemos únicamente las opciones de usuarios, hosts, grupos de ambos, permisos de *sudo* y configuración de *automount* para los directorios *HOME* de los usuarios.

Como demostración crearemos dos tipos de usuarios, uno de alumnos y otro de administradores. Los alumnos solo tendrán capacidad de acceso a las máquinas, mientras que los administradores tendrán permisos de ejecución de *sudo* para lanzar cualquier comando como usuario *root*. También crearemos dos tipos de hosts, uno para puestos de trabajo de alumnos, y otro para servidores de infraestructura, como por ejemplo los propios servidores de FreeIPA y de Foreman.

Para la creación de grupos acudiremos a la opción *Groups* dentro del menú *Identity*. Luego dentro del apartado de grupos de usuario pulsaremos el botón de añadir, y crearemos un nuevo grupo *alumnos*. Aunque opcional, proporcionaremos una descripción, y no especificaremos identificador de grupo (*GID*) para que se nos proporcione uno automáticamente. No crearemos grupo de administradores, ya que usaremos el grupo *admins* creado en la instalación de FreeIPA. Después, dentro del apartado de grupos de hosts crearemos dos grupos: uno llamado *puestos* y otro llamado *infraestructura*.



The screenshot shows a web-based form titled "Add User Group" with a close button (X) in the top right corner. The form contains the following fields and options:

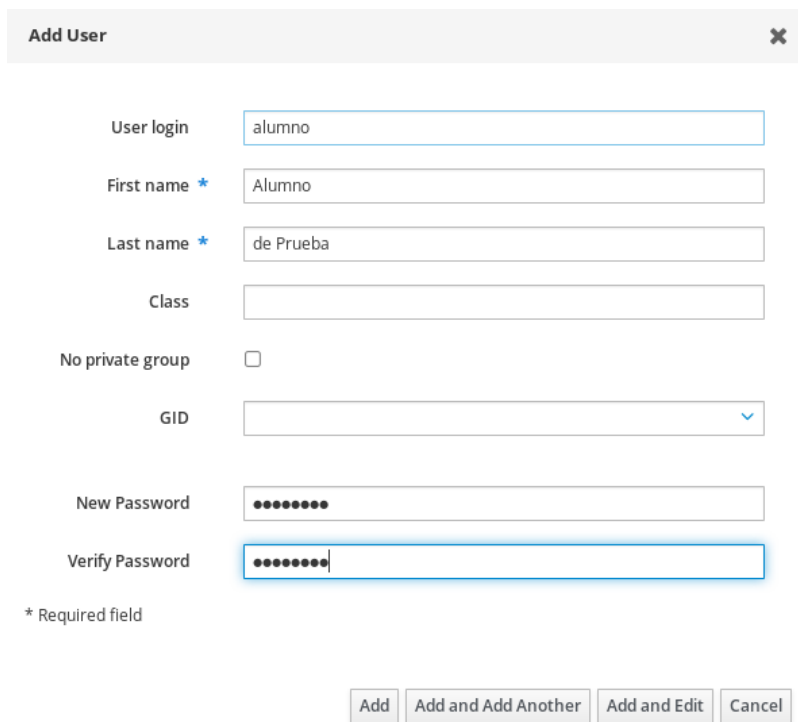
- Group name ***: A text input field containing the value "alumnos".
- Description**: A text area containing the value "Grupo para alumnos de la ETSISI".
- Group Type**: Three radio button options: "Non-POSIX", "External", and "POSIX". The "POSIX" option is selected.
- GID**: An empty text input field.

At the bottom left of the form, there is a note: "* Required field". At the bottom right, there are four buttons: "Add", "Add and Add Another", "Add and Edit", and "Cancel".

Figura 6: Creación de grupo de alumnos

A continuación, en el apartado *users*, crearemos dos usuarios, uno llamado *alumno* y otro llamado *profesor*. Para la creación proporcionaremos un nombre de acceso a las máquinas,

nombre y apellidos del usuario y contraseña inicial. Tras crear los usuarios, editaremos cada uno de ellos, y en el apartado *User Groups* añadiremos el grupo correspondiente a cada usuario: *alumnos* para el usuario *alumno*, y *admins* para el usuario *profesor*.



The screenshot shows a web form titled "Add User" with a close button (X) in the top right corner. The form contains the following fields and controls:

- User login:** Text input field containing "alumno".
- First name *:** Text input field containing "Alumno".
- Last name *:** Text input field containing "de Prueba".
- Class:** Text input field.
- No private group:** A checkbox that is currently unchecked.
- GID:** A dropdown menu.
- New Password:** Text input field with masked characters (dots).
- Verify Password:** Text input field with masked characters (dots).

Below the form, there is a legend: "* Required field". At the bottom of the form, there are four buttons: "Add", "Add and Add Another", "Add and Edit", and "Cancel".

Figura 7: Creación de usuario *alumno*

Tras esto proporcionaremos reglas de acceso y de ejecución a sus correspondientes grupos. Para ello, en el apartado *Policy*, accederemos a la sección *Host-Based Access Control*. Encontramos una regla *allow_all*, la cual deshabilitaremos, ya que queremos denegar el acceso de los alumnos a los servidores de infraestructura. Añadiremos una nueva regla *allow_alumnos*, y editaremos inmediatamente la misma para indicar que el grupo de usuarios *alumnos* podrá acceder a cualquier servicio del grupo de hosts *puestos*.

Who

User category the rule applies to: Anyone Specified Users and Groups

<input type="checkbox"/>	Users	Delete +Add
<input type="checkbox"/>	User Groups	Delete +Add
<input type="checkbox"/>	alumnos	

Accessing

Host category the rule applies to: Any Host Specified Hosts and Groups

<input type="checkbox"/>	Hosts	Delete +Add
<input type="checkbox"/>	Host Groups	Delete +Add
<input type="checkbox"/>	puestos	

Via Service

Service category the rule applies to: Any Service Specified Services and Groups

<input type="checkbox"/>	Services	Delete +Add
<input type="checkbox"/>	Service Groups	Delete +Add

Figura 8: Regla de HBAC para grupo de alumnos

De la misma forma crearemos una nueva regla *allow_admins* para permitir el acceso de los administradores a todos los equipos y servicios. Para poder probar esta regla tendremos que volver al apartado *Identity*, y en la sección de *Hosts* seleccionar el servidor de FreeIPA y añadirlo al grupo de hosts de *infraestructura*.

Adicionalmente, crearemos una regla de *sudo* para los administradores. Accederemos al apartado *Policy*, y en la sección *sudo* elegiremos la opción *Sudo Rules*. Crearemos una nueva regla *sudo_admins* y editaremos la misma para indicar las siguientes opciones: grupo de usuarios *admins*, cualquier host, cualquier comando, y posibilidad de ejecución como cualquier usuario o grupo. Aunque podríamos dar permisos de ejecución a ciertos comandos únicamente, permitimos todo, primero por simplicidad, y segundo porque el grupo de administradores debería ser capaz de ejecutar como usuario *root* cualquier comando.

Who

User category the rule applies to: Anyone Specified Users and Groups

<input type="checkbox"/> Users	External	Delete +Add
<input type="checkbox"/> User Groups		Delete +Add
<input type="checkbox"/> admins		

Access this host

Host category the rule applies to: Any Host Specified Hosts and Groups

<input type="checkbox"/> Hosts	External	Delete +Add
<input type="checkbox"/> Host Groups		Delete +Add

Run Commands

Command category the rule applies to: Any Command Specified Commands and Groups

Allow

<input type="checkbox"/> Sudo Allow Commands	Delete +Add
<input type="checkbox"/> Sudo Allow Command Groups	Delete +Add

Figura 9: Extracto de regla de *sudo* para grupo de administradores

Tras esto accederemos al servidor de FreeIPA para comprobar el correcto funcionamiento de la configuración que hemos creado:

```
# ssh alumno@freeipa
Password:
Connection closed by 192.168.100.2 port 22
#
# ssh profesor@freeipa
Password:
Last login: Sat Oct 13 18:32:03 2018 from gateway
Could not chdir to home directory /home/profesor: No such file or directory
-sh-4.2$ id
uid=1363400009(profesor) gid=1363400009(profesor) groups=1363400009(profesor),1363400000(admins)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
-sh-4.2$ sudo -l
[sudo] password for profesor:
Matching Defaults entries for profesor on freeipa:
!visiblepw, always_set_home, match_group_by_gid, env_reset, env_keep="COLORS DISPLAY
HOSTNAME HISTSIZE KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG
LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User profesor may run the following commands on freeipa:
(ALL : ALL) ALL
-sh-4.2$ █
```

Figura 10: Prueba de acceso con usuarios de FreeIPA

Podemos observar que al acceder como usuario *alumno* en el servidor *freeipa*, se cierra inmediatamente la conexión al no tener permiso de acceso. Por otro lado, al acceder como *profesor*, podemos entrar al servidor por la regla de HBAC *allow_admins*, y además ejecutar cualquier comando mediante *sudo*.

Vemos sin embargo un mensaje de error indicando que el directorio *HOME* del usuario no existe. Dado que vamos a servir por NFS los directorios *HOME*, tendremos que definir unas reglas de *Automount* para que cuando un usuario acceda a una máquina, se conecte su directorio *HOME* a la misma, y pueda disponer de sus datos en cualquier puesto de trabajo.

En primer lugar ejecutaremos el comando *ipa-client-automount* en el servidor de FreeIPA para configurar el servicio *automount* en el mismo. A continuación deberemos configurar dicho servicio, para lo que acudiremos al apartado *Network Services*, sección *Automount*. Seleccionaremos la ubicación *default*, y crearemos un nuevo *map* indirecto llamado *auto.userdirs*, que tenga como base el directorio */home*:

Add Automount Map ✕

Map Type Direct Indirect

Map *

Description

Indirect

Mount point *

Parent map

* Required field

Figura 11: Creación de *map* para directorios de usuario

Una vez creado el *map*, editaremos el mismo para añadir una nueva clave. La clave será ***, ya que queremos que se monte cualquier directorio bajo */home*. Ésta apuntará a la unidad de NFS con los directorios de usuario, utilizando una ruta acabada en */*, que referenciará a la propia clave:

Add Automount Key [X]

Key *

Mount *
information

* Required field

[Add] [Add and Add Another] [Add and Edit] [Cancel]

Figura 12: Creación de clave para directorios de usuario

De esta forma, cuando por ejemplo accedamos al directorio `/home/profesor`, se montará automáticamente en dicha ruta la unidad NFS `192.168.100.1:/userdirs/profesor`. Así solo se montarán de forma automática los directorios de los usuarios que conecten, en lugar de montar el directorio `/home` completo, lo que no es recomendable en términos de eficiencia y de seguridad.

Tras esto accederemos como el usuario `profesor`, y observaremos que ya no nos aparece el error anterior, y que se ha montado automáticamente su directorio de usuario:

```
# ssh profesor@freeipa
Password:
Last login: Mon Oct 15 21:59:41 2018 from gateway
-sh-4.2$ df -h .
Filesystem                Size  Used Avail Use% Mounted on
192.168.100.1:/userdirs/profesor 442G  45G 375G  11% /home/profesor
```

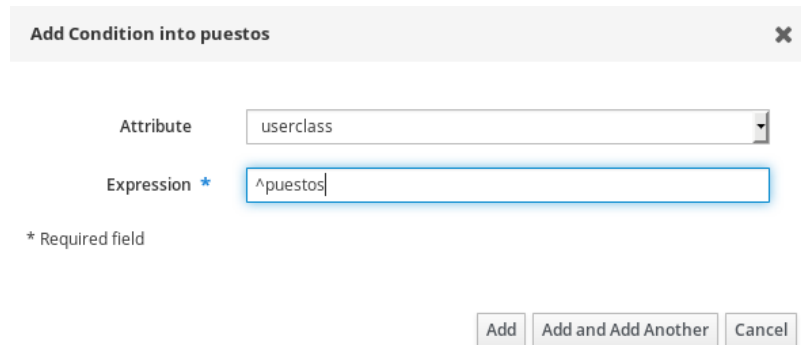
Figura 13: Acceso con usuario tras habilitar *automount*

Lo que tendremos que tener en cuenta es que los directorios de usuario deberán estar creados en el servidor NFS, con su propietario adecuado. Para evitar equivocaciones, sería posible automatizar mediante un único script la creación del usuario en el servidor de FreeIPA y su directorio en el servidor NFS, aunque en este ejemplo lo realizamos de forma manual por simplicidad.

Con esta configuración ya dispondríamos de un servidor de FreeIPA listo para proporcionar autenticación y autorización de usuarios a la infraestructura Linux propuesta. Sin embargo, vamos a realizar un par de retoques adicionales de cara al despliegue de puestos de trabajo de nuestra infraestructura.

Ya disponiendo del grupo de hosts *puestos*, vamos a crear una regla de automembresía, de forma que cuando se aprovisiona un nuevo puesto de trabajo, éste se añada automáticamente a dicho grupo para que los alumnos puedan acceder. Para ello accederemos al menú *Identity*, apartado *Automember*, y crearemos una nueva regla para el grupo *puestos*. Una vez creada la editaremos, y añadiremos una condición de tipo *Inclusive*, en la que se active la

automembresía si el atributo *userclass* del servidor empieza por la cadena *puestos*:



The image shows a dialog box titled "Add Condition into puestos" with a close button (X) in the top right corner. Inside the dialog, there are two main input fields. The first is labeled "Attribute" and is a dropdown menu currently showing "userclass". The second is labeled "Expression *" and is a text input field containing the text "^puestos". Below the "Expression *" field, there is a small asterisk and the text "* Required field". At the bottom of the dialog, there are three buttons: "Add", "Add and Add Another", and "Cancel".

Figura 14: Regla de automembresía para grupo de hosts *puestos*

Una vez creada la regla de automembresía, los nuevos puestos de trabajo desplegados podrán incluirse automáticamente en el grupo de hosts *puestos*. Lo siguiente será poder aprovisionar dichos puestos; pasaremos por tanto al apartado del servidor de provisión.

Apartado 4.3. Instalación de Foreman

De la misma forma que realizamos la instalación de FreeIPA, crearemos una nueva máquina virtual para el servidor de Foreman. Emplearemos también una imagen de CentOS 7.5, y asignaremos a la VM 2 CPUs virtuales, 4GB de RAM y un disco de 20GB.

Tras realizar la misma configuración inicial que en el servidor de FreeIPA, podremos habilitar la autenticación contra éste. Configuraremos como DNS de la máquina el servidor de FreeIPA, instalaremos el paquete *ipa-client*, y ejecutaremos los comandos *ipa-client-install* e *ipa-client-automount*. Al tener configurado FreeIPA como DNS, la obtención de datos para el registro es automática:

```

[root@foreman ~]# ipa-client-install
WARNING: ntpd time&date synchronization service will not be configured as
conflicting service (chronyd) is enabled
Use --force-ntpd option to disable it and force configuration of ntpd

Discovery was successful!
Client hostname: foreman.pfgupm.local
Realm: PFGUPM.LOCAL
DNS Domain: pfgupm.local
IPA Server: freeipa.pfgupm.local
BaseDN: dc=pfgupm,dc=local

Continue to configure the system with these values? [no]: yes
Skipping synchronizing time with NTP server.
User authorized to enroll computers: admin
Password for admin@PFGUPM.LOCAL:
Successfully retrieved CA cert
  Subject:   CN=Certificate Authority,0=PFGUPM.LOCAL
  Issuer:    CN=Certificate Authority,0=PFGUPM.LOCAL
  Valid From: 2018-10-08 23:55:49
  Valid Until: 2038-10-08 23:55:49

Enrolled in IPA realm PFGUPM.LOCAL
Created /etc/ipa/default.conf
New SSSD config will be created

```

Figura 15: Extracto de registro de Foreman en FreeIPA

A continuación procederemos a la instalación de Foreman siguiendo las instrucciones que aparecen en su manual [14]. Tenemos tres vías diferentes de instalación: mediante el instalador de Foreman (que utiliza Puppet), con paquetes RPM de la distribución, o a través del código fuente. Seleccionaremos la primera opción, ya que es la recomendada.

Para realizar la instalación, primero deberemos añadir los repositorios de Puppet, de EPEL (paquetes adicionales para distribuciones tipo Red Hat) y de Foreman (indicados en el manual). Una vez hecho esto podremos descargar el paquete *foreman-installer*, que utilizaremos para instalar Foreman. Como por defecto el instalador no habilita el DHCP y deseamos que sí esté en funcionamiento, incluiremos las opciones de instalación *foreman-proxy-dhcp* y *foreman-proxy-dhcp-range*. También, para el registro con FreeIPA que haremos más adelante, emplearemos las opciones *foreman-proxy-realm* y *foreman-proxy-realm-principal*:

```

[root@foreman ~]# foreman-installer --foreman-proxy-dhcp=true --foreman-proxy-dhcp-range="192.168.100.200 192.168.100.254" \
> --foreman-proxy-realm=true --foreman-proxy-realm-principal=realm-proxy@PFGUPM.LOCAL
Installing          Done          [100%] [.....]
Success!
* Foreman is running at https://foreman.pfgupm.local
  Initial credentials are admin / fGXV9td9qLYWqU9z
* Foreman Proxy is running at https://foreman.pfgupm.local:8443
* Puppetmaster is running at port 8140
The full log is at /var/log/foreman-installer/foreman.log

```

Figura 16: Opciones de instalación de Foreman

El rango de DHCP asignado podrá ser pequeño, ya que solo se usará para que las máquinas puedan arrancar por PXE durante el aprovisionamiento. Hay que tener en cuenta eso sí, que el tiempo de *lease* de DHCP por defecto es bastante elevado, así que es posible que con un rango pequeño, al instalar muchas máquinas en poco tiempo se llene la tabla de asignaciones del DHCP y haya que vaciarla a mano.

Con Foreman ya instalado, de nuevo tendremos que habilitar las reglas de firewall necesarias para permitir el acceso al servicio. En este caso deberemos habilitar los servicios *RH-Satellite-6* y *dhcp* usando el comando *firewall-cmd*, tal y como vimos en la instalación de FreeIPA.

Una vez permitido el tráfico, podremos acceder mediante HTTPS al servidor de Foreman. Tras autenticarnos con las credenciales proporcionadas durante la instalación, modificaremos la contraseña de administrador por otra a nuestra elección.

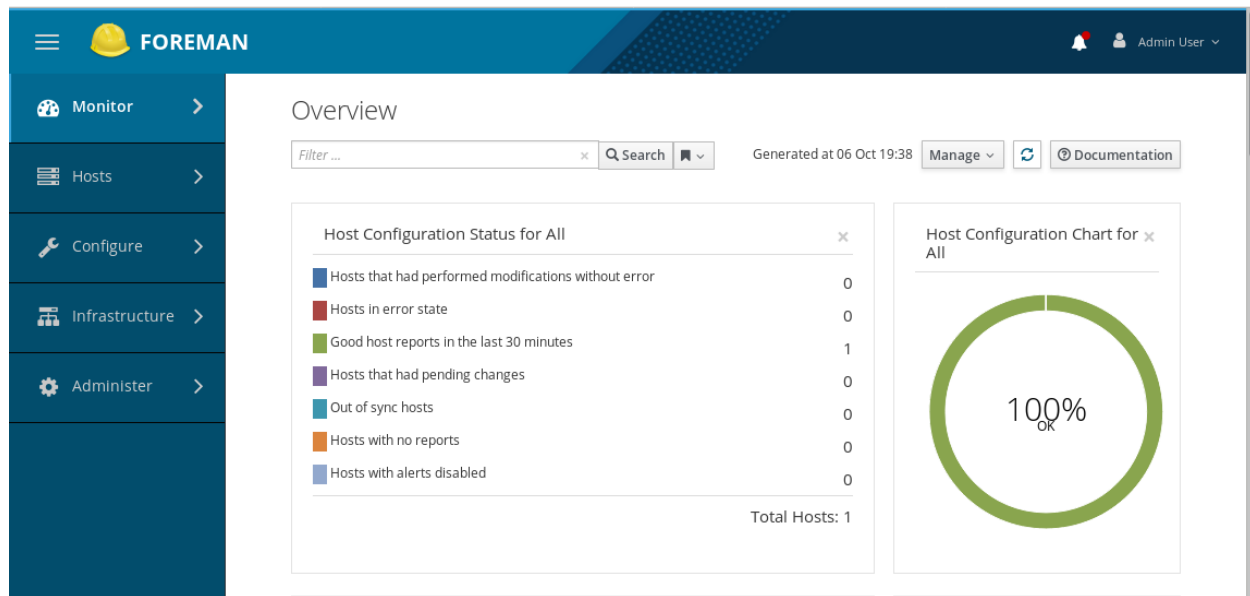


Figura 17: Pantalla inicial de Foreman

Podemos observar diferentes secciones dentro de la interfaz de Foreman. En rasgos generales su función es la siguiente:

- *Monitor*: muestra estadísticas, datos de hosts registrados, reportes, etc.
- *Hosts*: opciones de aprovisionamiento de máquinas.
- *Configure*: mayoritariamente para la configuración de máquinas vía Puppet.
- *Infrastructure*: datos a nivel de infraestructura como redes, plataformas de despliegue, etc.
- *Administer*: opciones de administración del propio Foreman.

Para poder aprovisionar máquinas, necesitamos configurar un perfil de sistema operativo. Foreman ya proporciona ciertos datos en la instalación base, como por ejemplo la URL para la instalación por red de CentOS, pero igualmente hay que configurar el perfil. En el menú de *Hosts* seleccionaremos *Operating System*, y crearemos un nuevo sistema operativo

CentOS 7.5.1804 con los detalles de nuestro sistema operativo (versión, familia, etcétera).

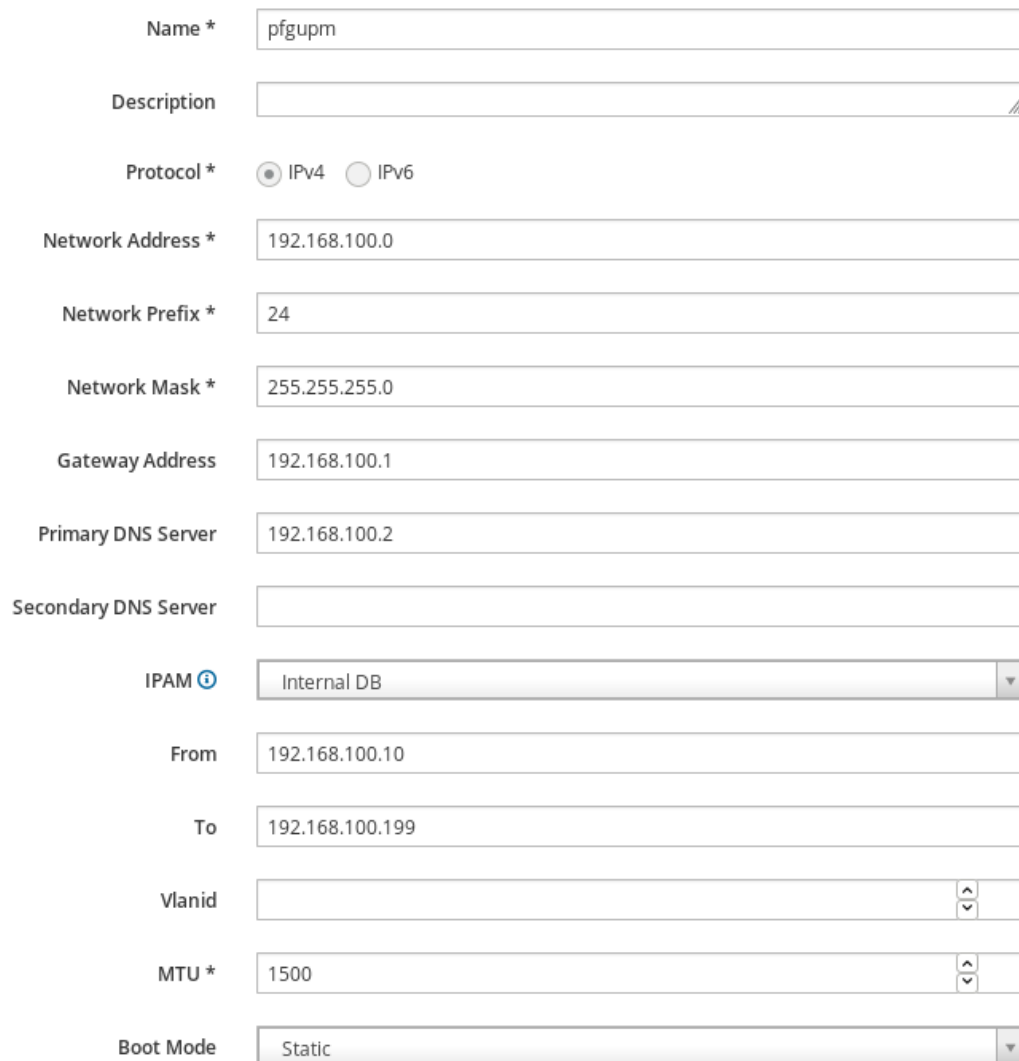
The screenshot shows the Foreman OS creation interface. It consists of several input fields and a list of architectures.

- Name ***: Input field containing "CentOS". Help text: "OS name from facter; e.g. RedHat".
- Major ***: Input field containing "7". Help text: "OS major version from facter; e.g. 6".
- Minor**: Input field containing "5". Help text: "OS minor version from facter; e.g. 5".
- Description**: Input field containing "CentOS 7.5.1804". Help text: "OS friendly name; e.g. RHEL 6.5".
- Family**: Dropdown menu showing "Red Hat".
- Root Password Hash**: Dropdown menu showing "SHA256". Help text: "Hash function to use. Change takes effect for new or updated hosts."
- Architectures**: A section with two panes:
 - All items**: A list containing "i386".
 - Selected Items**: A list containing "x86_64".

Figura 18: Datos de creación de sistema operativo en Foreman

Deberemos, en el apartado de tabla de particiones, asignar la tabla *Kickstart default*, y en el de medios de instalación el *CentOS mirror*. Después, tendremos que asociar plantillas de provisión a nuestro sistema operativo. Para ello, acudiremos al menú *Hosts*, y en el apartado *Provisioning Templates* seleccionaremos las plantillas *Kickstart default*, *Kickstart default PXELinux* y *Kickstart default PXEGrub2*, y las asociaremos al OS *CentOS 7.5.1804*. Tras esto, volveremos a editar nuestro sistema operativo, y en el apartado de *Templates* asociaremos las plantillas de provisión recién indicadas.

También, para poder realizar aprovisionamiento, tendremos que definir una red para las máquinas virtuales. Para ello, acudiremos al menú *Infrastructure*, seleccionaremos *Domains*, y crearemos un nuevo dominio *pfgupm.local*. Después, en el apartado *Subnets* crearemos una nueva subred asociada a este dominio. Le proporcionaremos los parámetros adecuados (como dirección de red, máscara de subred, etcétera), y seleccionaremos como método de IPAM la base de datos interna para que los puestos dispongan de IP fija. Indicaremos un rango de asignación de red, esta vez mayor, ya que supuestamente albergará bastantes equipos de trabajo, y finalmente le indicaremos que utilice método de carga estático.



The screenshot displays the 'Create Network' form in the Foreman web interface. The form is organized into several sections:

- Name ***: Text input field containing 'pfgupm'.
- Description**: Text area for additional information.
- Protocol ***: Radio buttons for 'IPv4' (selected) and 'IPv6'.
- Network Address ***: Text input field containing '192.168.100.0'.
- Network Prefix ***: Text input field containing '24'.
- Network Mask ***: Text input field containing '255.255.255.0'.
- Gateway Address**: Text input field containing '192.168.100.1'.
- Primary DNS Server**: Text input field containing '192.168.100.2'.
- Secondary DNS Server**: Empty text input field.
- IPAM**: Dropdown menu with 'Internal DB' selected.
- From**: Text input field containing '192.168.100.10'.
- To**: Text input field containing '192.168.100.199'.
- Vlanid**: Spinner control for VLAN ID.
- MTU ***: Spinner control with '1500' selected.
- Boot Mode**: Dropdown menu with 'Static' selected.

Figura 19: Creación de subred en Foreman

Tendremos además que asociar dicha subred a nuestro dominio *pfgupm.local*, y seleccionar como proxies DHCP y TFTP el propio servidor de Foreman. Esta configuración sería ya suficiente para aprovisionar equipos con Foreman. Sin embargo además queremos que nuestros equipos estén integrados con FreeIPA, para lo que deberemos realizar alguna configuración adicional.

En primer lugar deberemos ejecutar el script *foreman-prepare-realm*, tal y como se especifica en el manual de Foreman. Este script asignará automáticamente una serie de permisos al servidor de Foreman para crear y registrar equipos, y nos devolverá un fichero *keytab* de Kerberos, que proporcionará autenticación contra el servidor de FreeIPA. Posteriormente deberemos modificar los permisos de dicho fichero para que pueda ser empleado por los procesos de Foreman.

```

[root@foreman ~]# foreman-prepare-realm admin realm-proxy
Password for admin@PFGUPM.LOCAL:
-----
Added privilege "Smart Proxy Host Management"
-----
Privilege name: Smart Proxy Host Management
Description: Smart Proxy Host Management
-----
Added permission "Add Host Enrollment Password"
-----
Permission name: Add Host Enrollment Password
Granted rights: add
Effective attributes: userpassword
Bind rule type: permission
Subtree: cn=computers,cn=accounts,dc=pfgupm,dc=local
Type: host
Permission flags: SYSTEM, V2
Privilege name: Smart Proxy Host Management
Description: Smart Proxy Host Management
Permissions: Retrieve Certificates from the CA, System: Add DNS Entries,
System: Read DNS Entries, System: Remove DNS Entries, System:
Update DNS Entries, System: Manage Host Certificates, System:
Manage Host Enrollment Password, System: Manage Host Keytab,
System: Modify Hosts, System: Remove Hosts, System: Manage
Service Keytab, System: Modify Services, Add Host Enrollment
Password
-----
Number of permissions added 13
-----

```

Figura 20: Extracto de script de integración de Foreman y FreeIPA

Tras esto crearemos un nuevo reino de Kerberos en Foreman. Accederemos al menú *Infrastructure*, y aquí en el apartado *Realms*, donde crearemos un nuevo reino *PFGUPM.LOCAL*:

The screenshot shows a form with three fields:

- Name ***: A text input field containing "PFGUPM.LOCAL".
- Realm Type ***: A dropdown menu with "FreeIPA" selected.
- Realm proxy ⓘ ***: A dropdown menu with "foreman.pfgupm.local" selected.

Figura 21: Creación de reino de Kerberos en Foreman

Para configurar también correctamente el servicio de *Automount* debemos especificar los parámetros *freeipa_automount* y *freeipa_mkhome_dir*, que adoptarán valores *true* y *false* respectivamente. Los parámetros en Foreman pueden especificarse en múltiples niveles (a nivel global, de dominio, de subred... y así hasta un único equipo) y podrán irse heredando o sobrescribiendo de un nivel a otro. En esta instalación asignaremos dicho parámetro a nivel de grupo de hosts.

Un grupo de hosts consiste en un conjunto de equipos que comparte una serie de parámetros comunes de cara a la provisión. De esta forma no tendremos que especificar sistema operativo, plantillas de provisión, etcétera, cada vez que queramos instalar un nuevo equipo. Crearemos un nuevo grupo desde el menú *Configure*, apartado *Host Groups*, el

cual llamaremos *puestos* para que coincida con la regla de automembresía que creamos en FreeIPA.

Dentro de dicho grupo, además del nombre, configuraremos los datos de los apartados *Network*, *Operating System* y *Parameters*, ya que el resto de la configuración que aparece corresponde a Puppet, que no estamos utilizando. Generalmente para cada opción se nos proporcionará una lista desplegable, que contendrá un único valor, ya que solo hemos configurado un elemento por cada tipo de recurso. Además configuraremos los parámetros de FreeIPA indicados anteriormente:

Host Group	Puppet Classes	Network	Operating System	Parameters
			Architecture	x86_64
			Operating system *	CentOS 7.5.1804
			Media *	CentOS mirror
			Partition Table *	Kickstart default
			PXE loader ⓘ	PXELinux BIOS
			Root Pass	*****

Figura 22: Configuración de sistema operativo para grupo de hosts *puestos*

En este punto ya estaríamos en condiciones de desplegar un nuevo puesto de trabajo integrado con FreeIPA. Se podrían hacer más personalizaciones, como por ejemplo modificar la plantilla de aprovisionamiento de Kickstart para ampliar la paquetería de los sistemas desplegados, y añadir así un entorno de escritorio o utilidades necesarias para los alumnos. De nuevo omitiremos estas personalizaciones para mantener un entorno de pruebas sencillo y desplegar equipos con un software mínimo, procedimiento que abarcaremos en el siguiente apartado.

Apartado 4.4. Instalación de puesto de trabajo

Vamos a proceder finalmente a la creación de un nuevo puesto de trabajo. Para ello crearemos una nueva máquina virtual con 2 CPUs y 2GB de RAM (esto último es importante, ya que el despliegue falla con menos memoria), pero no le asignaremos ninguna imagen de instalación, sino que configuraremos la VM para que arranque a través de la tarjeta de red mediante PXE. Para el registro necesitaremos obtener la dirección MAC de la tarjeta de red, ya que será el dato que utilice Foreman para identificar al equipo:

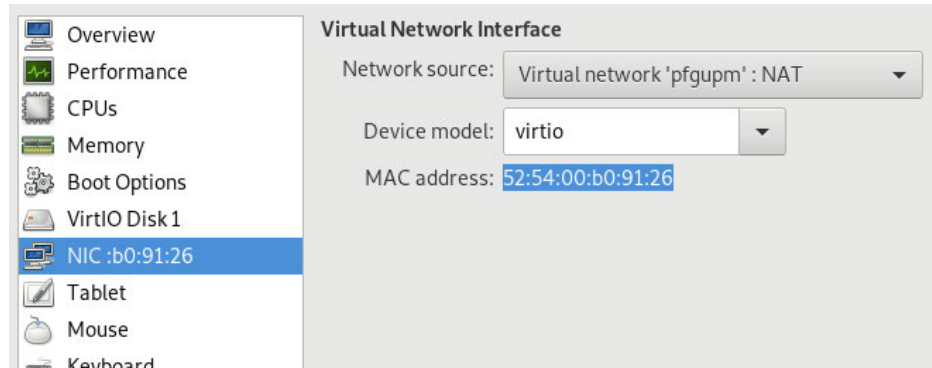


Figura 23: Dirección MAC de la tarjeta de red

Una vez realizada la configuración inicial de la máquina virtual, procederemos al registro de ésta en Foreman. Dentro de *Hosts* accederemos a *Create Host*, y crearemos un nuevo equipo *puesto1*. Seleccionaremos el grupo de hosts *puestos*, con lo que ya tendremos configuradas la mayor parte de las opciones de provisión, y luego accederemos al apartado *Interfaces*, donde editaremos la configuración de la interfaz de red para asociarla a la dirección MAC de nuestra VM:

The image shows a web form for configuring a network interface in Foreman. The form includes the following fields and options:

- Type:** A dropdown menu set to "Interface".
- MAC Address:** A text input field containing "52:54:00:b0:91:26".
- Device Identifier:** An empty text input field.
- DNS Name:** A text input field containing "puesto1".
- Domain:** A dropdown menu set to "pfgupm.local".
- IPv4 Subnet:** A dropdown menu set to "pfgupm (192.168.100.0/24)".
- IPv6 Subnet:** A dropdown menu set to "No subnets".
- IPv4 Address:** A text input field containing "192.168.100.10". Below this field is a blue link labeled "Suggest new".
- IPv6 Address:** An empty text input field.
- Managed:** A checkbox that is checked.
- Primary:** A checkbox that is checked.
- Provision:** A checkbox that is checked.
- Virtual NIC:** An unchecked checkbox.

Figura 24: Configuración de interfaz de red de Foreman

Tras esto únicamente tendremos que arrancar la máquina virtual y automáticamente conectará con el servidor de Foreman, el cual le proporcionará una imagen de CentOS, con lo que se comenzará con la instalación del equipo. A continuación podemos ver la pantalla inicial con el arranque por red de la máquina virtual:

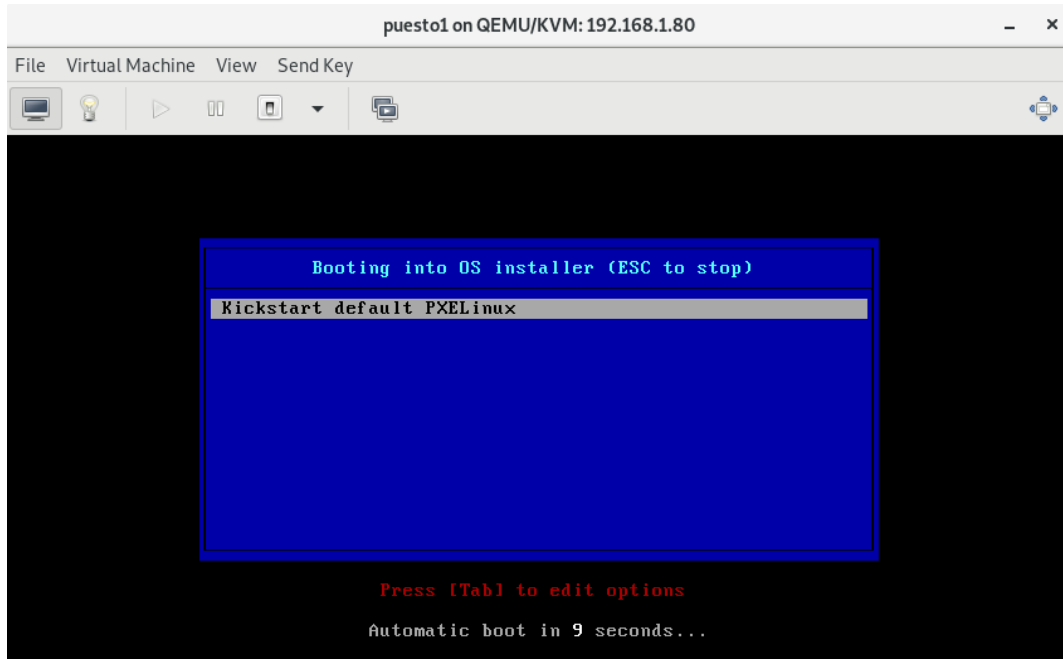


Figura 25: Arranque por red mediante PXE

En este proceso no solo se realizará la instalación básica del sistema operativo, sino que también se ejecutarán las tareas de post-instalación definidas en la plantilla de Kickstart seleccionada, como la actualización de la paquetería del equipo a la última versión disponible en los repositorios públicos y el registro contra el servidor de FreeIPA. Pasada media hora aproximadamente, tendremos instalado y accesible nuestro nuevo puesto de trabajo:

```
[root@foreman ~]# host puesto1.pfgupm.local
puesto1.pfgupm.local has address 192.168.100.11
[root@foreman ~]# ssh alumno@puesto1.pfgupm.local
Password:
Last login: Tue Oct 23 14:32:03 2018 from 192.168.100.3
-sh-4.2$ id -u
1363400008
-sh-4.2$ df .
Filesystem            1K-blocks      Used Available Use% Mounted on
192.168.100.1:/userdirs/alumno 463002624 50045952 389367808  12% /home/alumno
-sh-4.2$
-sh-4.2$ su - profesor
Password:
Last login: Tue Oct 23 14:31:35 UTC 2018 from 192.168.100.3 on pts/0
-sh-4.2$ sudo -l
[sudo] password for profesor:
Matching Defaults entries for profesor on puesto1:
!visiblepw, always_set_home, match_group_by_gid, env_reset, env_keep="COLORS DISPLAY
HOSTNAME HISTSIZE KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG
LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User profesor may run the following commands on puesto1:
(ALL : ALL) ALL
```

Figura 26: Comprobación de correcto funcionamiento de VM aprovisionada

En la anterior captura de pantalla podemos observar varios detalles: podemos ver que

se puede resolver mediante DNS el nombre *puesto1.pfgupm.local*, con lo que el registro en FreeIPA genera correctamente las entradas de DNS asociadas al equipo. Podemos conectar mediante SSH con el usuario *alumno*, con lo que la regla de HBAC para permitir el acceso de los alumnos a los puestos de trabajo funciona correctamente, y además se monta automáticamente mediante NFS su directorio de usuario. Por último, podemos acceder como usuario *profesor* y comprobar que las reglas de *sudo* se reciben correctamente y que dicho usuario puede ejecutar cualquier comando como superusuario.

Con estas comprobaciones hemos podido verificar que todas las funcionalidades de FreeIPA funcionan correctamente: autenticación y autorización (tanto de acceso como de *sudo*), reglas de automembresía del equipo, y sistema de *automount*. Ya dispondríamos por tanto de un equipo de trabajo, al que cualquier alumno registrado en la plataforma podría acceder, y en el que tendría acceso a sus datos de usuario al estar disponibles en una unidad de red NFS.

Por último, nos gustaría indicar una aclaración con respecto al aprovisionamiento: si el equipo fuera reiniciado volvería a tratar de arrancar primero mediante PXE y luego por unidad de disco local. Al arrancar por PXE no volvería a realizarse la instalación, ya que Foreman identificaría al equipo por su dirección MAC y detectaría que ya se encuentra instalado. Éste, por tanto, proporcionaría la instrucción de ceder el arranque al siguiente elemento de la cadena: el disco duro local con nuestra instalación inicial de Foreman.

Apartado 4.5. Diagrama de secuencia de aprovisionamiento

A continuación incluimos un diagrama de secuencia con el proceso de aprovisionamiento de un puesto de trabajo:

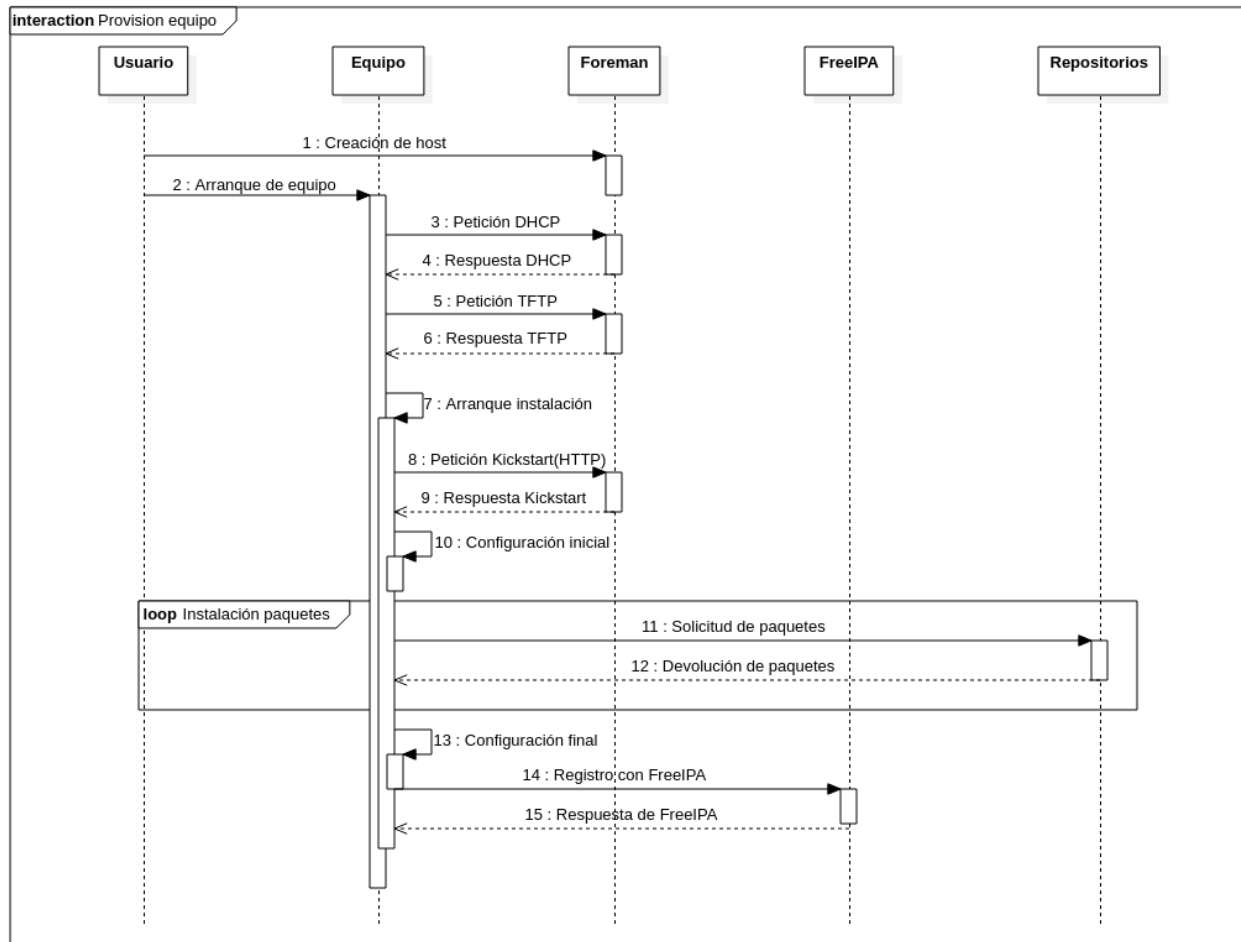


Figura 27: Diagrama de secuencia de aprovisionamiento

Capítulo 5. Compatibilidad y alternativas de software

Hemos visto ya cómo gestionar e instalar una infraestructura de puestos de trabajo basada en Linux. Sin embargo, hemos realizado una instalación mínima de sistema operativo, mientras que los puestos de trabajo Windows poseerán una amplia colección de software para el trabajo del día a día de los alumnos.

En esta sección del documento analizaremos varios programas encontrados en un centro de trabajo universitario, e indicaremos cuáles podremos ejecutar en Linux y cuáles no. Para los programas que no pudieran ser ejecutados en Linux, trataremos de proporcionar alternativas de código abierto, o también una serie de directrices para poder ejecutar este software bajo Linux.

Separaremos dicho software en diferentes categorías, comenzando por programas de propósito general, y continuando luego por herramientas de desarrollo informático, o más específicas de otras materias, como por ejemplo herramientas matemáticas. No se incluirán referencias bibliográficas en este apartado, sino que será en el anexo 2 de este documento donde detallaremos las URLs en donde hemos obtenido la información de compatibilidad de los programas en cuestión, así como las páginas principales de otras recomendaciones propuestas.

Al final de esta sección se encontrará una tabla a modo de resumen de los diferentes programas analizados y su compatibilidad con Linux.

Apartado 5.1. Software de propósito general

En primer lugar comenzaremos con utilidades típicas de un sistema de escritorio, que podemos encontrar no solo en entornos de trabajo, sino también en equipos personales.

En primer lugar hablaremos de navegadores típicos: Mozilla Firefox, Google Chrome e Internet Explorer. Firefox y Chrome podemos encontrarlos disponibles para Linux, aunque en el caso de Chrome recomendaríamos la alternativa libre Chromium (en la que se basa el primero). Internet Explorer ha sido el navegador por defecto de Microsoft Windows durante mucho tiempo, y por ello no es de extrañar que éste no se encuentre disponible para Linux. Sin embargo, teniendo disponibles las dos primeras opciones, creemos que Internet Explorer no sería realmente necesario para el entorno que se quiere proponer.

Sobre suites de productividad encontramos dos productos conocidos: Microsoft Office y LibreOffice. Éste último es la suite ofimática por excelencia del mundo Linux, y un reemplazo bastante aceptable para Microsoft Office. Acerca de la suite de Microsoft, ésta no se encuentra disponible para Linux, pero como alternativa para cuando LibreOffice no sea suficiente podremos utilizar Office Online, ya que es posible emplearlo a través

de cualquier navegador compatible, como Firefox o Chrome. De todos modos, los problemas que presenta LibreOffice suelen aparecer al tratar documentos de Microsoft Office. En un entorno en que solo se emplee LibreOffice no encontraríamos dichos problemas.

En cuanto a editores de texto encontramos típicamente los siguientes: Notepad, Wordpad y Notepad++. Notepad es un editor de texto sencillo incluido con Windows, el cuál podríamos reemplazar por el editor Gedit, incluido con el entorno de escritorio GNOME. Notepad++ es un editor de texto más completo que Notepad y Gedit, que aun siendo libre no está disponible para Linux. Como alternativa más directa encontramos NotepadQQ, muy similar al anterior. Wordpad se trata de un paso intermedio entre Notepad y Microsoft Word, y realmente podemos suplir su función con alguna de las alternativas para Linux mencionadas. En cualquier caso, Linux posee multitud de editores de texto, tanto en entorno de ventanas como de consola: Vim, Emacs, Nano... Podrían disponerse todos ellos a los usuarios, de forma que cada uno pueda elegir el editor más adecuado según la necesidad.

Como visor de documentos PDF encontramos el programa PDF-XChange Viewer. Éste no se encuentra disponible para Linux, pero disponemos de dos alternativas: en primer lugar utilizar el visor oficial (Adobe Reader). El problema de Adobe Reader es que solo está disponible para Linux hasta su versión 9, no siendo así las posteriores. En segundo lugar disponemos del visor Evince, perteneciente también al grupo de aplicaciones incluidas con GNOME.

Acerca de software de compresión encontramos 7-zip, el cual es libre pero no está disponible para Linux. Sin embargo, al tratarse de un formato libre, hay implementaciones también libres para Linux, como por ejemplo *p7zip*. Además, en Linux se incluyen diversas utilidades de compresión libres, como *gzip*, *bzip2* y *xz*, que combinadas con el archivero *tar* constituyen el principal método de compresión en entornos Linux. Acerca del formato ZIP, al igual que en Windows, se incluye integración directa en el entorno de escritorio GNOME.

Por último hablaremos de VLC como software de reproducción de vídeo, y de GIMP como software de edición de imágenes. En ambos casos estamos hablando de software libre, que está disponible además para Linux. De hecho en caso de haber encontrado otro software, como por ejemplo el reproductor de Windows Media y Adobe Photoshop, se habrían recomendado como alternativas los primeros.

Apartado 5.2. Entornos de desarrollo

En el ámbito informático es de gran importancia disponer de entornos de desarrollo, y es por tanto que trabajando sobre Linux deberemos disponer de programas de este tipo. A continuación estudiaremos los diferentes entornos que hemos encontrado, y si pueden ser utilizados o no bajo Linux.

En primer lugar encontramos muchos entornos de desarrollo programados sobre Java, que al tratarse de un entorno de ejecución multiplataforma funcionan también sobre Linux.

Éste es el caso por ejemplo de entornos populares como Eclipse, Netbeans, IntelliJ IDEA y JDeveloper.

Podemos encontrar otros ejemplos menos populares, pero utilizados en facultades, tales como Code::Blocks, GNAT Programming Studio o Kojo. Éstos, aunque se encuentren desarrollados bajo otros lenguajes, encontramos que también son compatibles con Linux.

Finalmente tenemos un caso particular, que es el de Visual Studio. Existe un entorno específico llamado Visual Studio Code que puede ser ejecutado bajo Linux. Sin embargo este programa solo está destinado a la edición de código para varios lenguajes (incluyendo lenguajes desarrollados por Microsoft como C# o F#), y no permite la compilación de programas como el resto de entornos mentados. Como alternativas podremos, o bien ejecutar Visual Studio bajo Windows a través de una máquina virtual, o bien emplear los servicios de nube de Azure DevOps para compilar el código.

Apartado 5.3. Entornos de ejecución

Además de los entornos de desarrollo, encontramos que para ciertos lenguajes necesitamos un entorno de ejecución, puesto que dichos lenguajes no producen código compilado. En su lugar dichos lenguajes generan código intermedio, que luego es ejecutado mediante una máquina virtual (como Java, por ejemplo), o bien su código directamente es interpretado y ejecutado por el entorno de ejecución. En esta sección trataremos dichos entornos.

Primero encontramos la familia de productos Java, que es el entorno de ejecución multiplataforma por excelencia. La implementación oficial de Oracle de los entornos de ejecución (JRE) y desarrollo (JDK) están por tanto disponibles bajo Linux. Aparte, existe una implementación libre (mantenida también por Oracle) llamada OpenJDK, que suele estar disponible de forma directa en los repositorios de la mayoría de las distribuciones Linux, incluyendo CentOS.

Como utilidad relacionada con Java hemos añadido en esta sección JavaCC, que es un generador de analizadores sintácticos. JavaCC no deja de ser más que un programa Java en formato JAR, y funciona correctamente sobre Linux.

Otros entornos de ejecución que tienen compatibilidad con Linux son FreePascal y SWI-Prolog (para los lenguajes Pascal y Prolog respectivamente, como es de suponer). Entornos que no funcionan bajo Linux serían Camlwin y WinGHCi Haskell, para los lenguajes Caml y Haskell. Sin embargo Camlwin es un entorno de ejecución que se encuentra anticuado, y la versión más actual OCaml sí está disponible para Linux. Por otro lado, WinGHCi no deja de ser un editor gráfico para el compilador GHC de Haskell, compilador que también está disponible para Linux.

Para el caso del simulador de ejecución de instrucciones WinMIPS64, como podemos deducir por su nombre es un programa nativo de Windows, el cual no funciona en Linux.

Existe sin embargo una alternativa, llamada EduMIPS64, que se trata de una adaptación de WinMIPS64 programada en Java.

Finalmente tenemos el compilador de ensamblador de Microsoft MASM (Microsoft Macro Assembler), que es propietario de Microsoft y exclusivo de Windows. Para Linux existen diferentes alternativas libres, siendo las principales el ensamblador de GNU (as) y el Netwide Assembler o NASM.

Apartado 5.4. Programación de hardware

Hemos querido separar en un apartado adicional los entornos de desarrollo de hardware programable. En primer lugar tenemos las suites de desarrollo de Xilinx: ISE y su sucesor Vivado. Ambas suites están disponibles para Linux, y permiten realizar desde la codificación y simulación de programas, hasta la programación de la FPGA.

Tenemos también el simulador de código VHDL ModelSim, el cual está disponible solo para Windows. Por otro lado, ModelSim funciona correctamente en Linux mediante Wine (utilizado para ejecutar programas nativos de Windows), o también puede ser ejecutado mediante una VM Windows. Otra alternativa sería emplear el simulador incluido con las suites de Xilinx.

Otro simulador de VHDL que podemos encontrar es el Veribest VHDL simulator, también disponible únicamente en Windows. Aquí indicaremos que dicho simulador se encuentra ya bastante obsoleto, y que se recomendaría emplear uno de los otros dos simuladores mencionados.

Apartado 5.5. Otras utilidades de diseño y desarrollo

A continuación pasaremos a discutir acerca de otras utilidades relacionadas con el diseño y el desarrollo, como por ejemplo sistemas de control de versiones, herramientas de modelado, y en general programas que no entrarían en otra de las categorías definidas, y para los que no hemos considerado necesario generar una categoría propia.

En primer lugar encontramos la utilidad Git. Por lo que hemos observado, Git es empleado en Windows a través de un entorno conocido como Cygwin, que es una simulación de un intérprete de comandos tipo Linux. Por tanto, el uso de Git bajo el terminal de Linux nos resultará más natural que tener que ejecutarlo mediante un simulador. Comentar además que Git fue desarrollado originalmente por Linus Torvalds, creador del sistema operativo Linux.

Tenemos también utilidades de edición de ficheros con un formato específico, que no son tan complejas como para tratarse de entornos de desarrollo completos, pero que tampoco podrían ser consideradas utilidades de propósito general. Es el caso por ejemplo de XML

Copy Editor (para edición de ficheros XML), y de HTML-Kit Tools (para HTML). XML Copy Editor posee compatibilidad con Linux, pero HTML-Kit Tools no, aunque en la propia web del producto se indica que sí puede funcionar mediante Wine.

Encontramos también herramientas de modelado en UML, como por ejemplo StarUML y ArgoUML, ambas compatibles con Linux. También tenemos la herramienta de gestión de requisitos Rational DOORS de IBM. Ésta posee arquitectura cliente/servidor, y solo la parte servidor puede ser ejecutada bajo Linux. Para la parte cliente será necesaria su ejecución con algún tipo de emulación, o bien mediante Wine, o bien con una máquina virtual Windows.

Otras utilidades que no podíamos cuadrar en otros apartados son el editor de diagramas yEd Graph Editor, y el sistema de información geográfica gvSIG Desktop. Afortunadamente, ambos también disponen de versión de acceso libre para el sistema operativo Linux.

Por último tenemos herramientas relacionadas con Big Data. Es el caso por ejemplo de Weka, utilidad para la minería de datos y aprendizaje automático. Éste es un programa libre y escrito en Java, y por tanto puede ser ejecutado bajo Linux. También es el caso de Microsoft Power BI Desktop, herramienta de visualización de datos y generación de informes. Power BI no está disponible para Linux, pero sí podría ser ejecutado bajo VM. También tenemos herramientas similares que sí funcionan en Linux, como por ejemplo Kibana.

Apartado 5.6. Herramientas de redes

En esta sección encontramos utilidades de conexión remota, transferencia de ficheros, o también herramientas de análisis y diseño de red. Empezamos hablando de programas como Putty o WinSCP, utilidades libres para la conexión y copia de ficheros a través de protocolo SSH. Ambos casos son implementaciones para Windows de dicho protocolo, que es usado principalmente en sistemas Linux/UNIX, con lo que no habría problema en encontrar utilidades similares, como las de la implementación OpenSSH.

Para otros usos de Putty, como sesiones *telnet* o conexiones por cable de consola, tenemos las utilidades *telnet/nc/ncat* para el primero, o bien *minicom* para el segundo. En cuanto a WinSCP nos permite transferencias por SCP/SFTP (basado en SSH), pero también FTP y FTPS. Para las primeras tenemos las utilidades *scp/sftp* del paquete OpenSSH, y para lo segundo podemos utilizar o bien algún cliente tipo *lftp*, o también mediante *cURL*.

En cuanto a herramientas de análisis de red, la principal que encontramos en entornos Windows es Wireshark, también disponible para Linux. En Linux además tenemos *tcpdump* como herramienta de captura, que es ampliamente utilizado en entornos de servidor para realizar capturas desde línea de comando. Las capturas obtenidas con *tcpdump* pueden ser además analizadas mediante Wireshark.

El último software de estas características observado es OPNet Modeler, una herramienta

de diseño y simulación de redes de comunicaciones. Hay que indicar primero que dicho software ya no existe como tal, sino que su denominación ha pasado a ser Riverbed Modeler. En cualquier caso, este software está disponible para Linux.

Finalmente queremos hablar acerca de utilidades de conexión remota en modo gráfico. En Windows es habitual utilizar Terminal Services, basado en protocolo RDP (propietario). En Linux disponemos de implementaciones libres de dicho protocolo, tanto del lado de cliente como de servidor. Sin embargo, lo habitual en entornos Linux es el uso de utilidades de conexión remota que empleen protocolo VNC, que aunque tenga implementaciones de uso comercial se trata de un protocolo libre.

Apartado 5.7. Herramientas de seguridad

Vamos a pasar a analizar diferentes herramientas relacionadas con la seguridad, como por ejemplo software de gestión de certificados, utilidades criptográficas, etcétera. En primer lugar comentar que se ha observado en el entorno de trabajo Windows la suite de utilidades y librerías de OpenSSL. Como su nombre nos hace suponer, OpenSSL es un software libre y no solo está disponible para Linux, sino que habitualmente se encuentra incluido en la instalación del sistema operativo, incluso en los modos de instalación más reducidos.

Hemos encontrado también la utilidad de gestión de certificados X - Certificate and Key management (o XCA). Este programa está disponible para Linux, pero la utilidad más típica para gestión de certificados bajo Linux es OpenSSL. Si bien es cierto que su uso resulta algo complejo, también resulta más didáctico por otro lado. También existen scripts de ayuda, como `easy-rsa` (incluido con OpenVPN), que nos permiten gestionar de forma sencilla una infraestructura de CA.

Por último tenemos PGP, un software de cifrado, descifrado y firma digital que posee licencia privativa. PGP se considera ya obsoleto, y ha sido reemplazado por el formato de mensajes OpenPGP, para el que disponemos una implementación libre bajo Linux conocida como GPG.

Queremos indicar también que en el entorno de trabajo estudiado, se han encontrado muchas utilidades desarrolladas por la propia facultad para el estudio y el aprendizaje de algoritmos de hashing, cifrado y demás. Sobre estos programas, encontraremos una serie de directrices de ejecución generales en el apartado 5.11 de este documento, ya que no hemos podido analizarlos de forma individual.

Apartado 5.8. Bases de datos y almacenamiento

En este apartado se discutirá acerca de la compatibilidad de software de bases de datos, y de otros programas relacionado con el almacenamiento que no hayamos podido incluir en otra categoría. Vamos adelantando que todos los casos encontrados van a ser compatibles

con Linux, ya que casi todos los sistemas gestores de bases de datos pueden y suelen ser ejecutados bajo este sistema operativo, salvo algunas excepciones muy concretas como Microsoft SQL Server.

Sistemas bien conocidos como Oracle Database, MySQL (y la herramienta MySQL Workbench) o PostgreSQL, pueden ser ejecutados bajo Linux. En cuanto a MySQL, nos gustaría indicar que la comunidad se está decantando más por su variante MariaDB, ya que MySQL fue adquirido por Oracle y posee componentes de licencia comercial, mientras que MariaDB es completamente libre. Recomendaríamos por tanto el uso de este último para el entorno Linux que proponemos.

BaseX es una BD basada en XML menos conocida, pero también libre y disponible para Linux. Sin embargo, debido a la existencia de sistemas gestores de BD más populares o al uso cada vez más reducido de XML para almacenar información (en favor de otros formatos como JSON), recomendaríamos usar otro sistema como los indicados en el párrafo anterior, u otras opciones más modernas como MongoDB.

Hemos decidido incluir en este apartado el paquete XAMPP, que se trata de un conjunto de software dedicado a aplicaciones web multiplataforma, el cual incluye una BD MariaDB (antes MySQL), un servidor web Apache, y entornos de ejecución de PHP y Perl. Como indicamos, es multiplataforma y puede correr bajo diferentes sistemas operativos, Linux incluido.

Así mismo, hemos añadido aquí también Microsoft Azure Storage Explorer, que no se trata realmente de un sistema de almacenamiento, sino de un visualizador y gestor de datos almacenados en el servicio de nube de Azure Storage, el cual sí es un sistema de almacenamiento/base de datos. Este programa cliente, aunque se trate de un software creado por Microsoft, se encuentra también disponible para Linux.

Apartado 5.9. Software de virtualización

Sobre lo relacionado al software de virtualización, el que parece estar más presente es Oracle VM VirtualBox, programa de virtualización libre que también está disponible en Linux. También recomendaríamos el uso de KVM, que es el hipervisor que utilizan actualmente muchos productos de virtualización basados en Linux (como oVirt u OpenStack), y que proporciona un rendimiento mayor que VirtualBox [15].

En el entorno Windows analizado también hemos encontrado Vagrant para el aprovisionamiento de máquinas virtuales. Este software está disponible también para Linux, y si bien acepta la creación de VMs bajo KVM, la gran parte de desarrollos de Vagrant utilizan VirtualBox. Recomendaríamos por tanto el uso de KVM para la virtualización en general de VMs dado su mayor rendimiento, y adicionalmente el uso de VirtualBox para los casos concretos en que se vaya a usar Vagrant.

También tenemos el caso de Docker, que se trata de una virtualización más ligera basada en contenedores. Docker se suele emplear de hecho desde sistemas anfitriones Linux, ya que los huéspedes (Linux también), aunque poseen sus propias librerías y utilidades, hacen uso del kernel del sistema anfitrión. Docker para Windows requiere de un motor de interpretación intermedio, con lo que no solo es posible usar Docker bajo Linux, sino que es más recomendable que hacerlo sobre Windows.

Apartado 5.10. Software matemático

En la última categoría discutida hablaremos de utilidades matemáticas. Indicaremos rápidamente que programas encontrados tales como Matlab, Maple o wxMaxima, están disponibles para su uso bajo Linux. Por otro lado, programas como Derive o Statgraphics solo disponen de versión para Windows, con lo que no sería posible su ejecución en Linux de forma nativa.

En cualquier caso, Derive se trata de un programa ya anticuado, así que recomendaríamos la búsqueda de alguna alternativa, y ya de paso que además funcionara con Linux. Sin embargo, dado nuestro desconocimiento en dichas utilidades, no queremos aventurarnos a proporcionar otras alternativas, aunque sí indicaremos que en caso de no poder encontrar ninguna, podemos tratar de ejecutar estos programas mediante Wine o a través de una máquina virtual Windows.

Apartado 5.11. Directrices generales de ejecución de software sobre Linux

Finalmente, queríamos proporcionar una serie de consideraciones generales y directrices para la ejecución de software sobre Linux, aunque ya habrán podido ser inferidas por lo discutido en los anteriores apartados. Creemos que éstas resultarán útiles a la hora de ejecutar ciertos programas, como por ejemplo utilidades desarrolladas por antiguos alumnos de la universidad, de las que seguramente no se disponga del código fuente, y que no esté contemplada su portabilidad a Linux.

En primer lugar indicaremos que muchas utilidades actualmente están desarrolladas en Java, y que al tratarse de un lenguaje interpretado multiplataforma podremos ejecutar dichas utilidades bajo Linux mediante el entorno de ejecución de Java. Éste ha sido el caso de gran parte de las utilidades discutidas en esta sección del documento.

En segundo lugar, en Linux disponemos de Wine, una capa de compatibilidad mediante la que podremos ejecutar bajo Linux ciertos programas compilados para Windows. Junto con Wine disponemos también de Mono, que es una implementación de código abierto del Framework .NET de Microsoft. Gracias a estos dos proyectos es posible la ejecución sobre Linux de código nativo de Windows, aunque bien es cierto que solo ciertos programas poseerán el 100% de funcionalidad. En la web de Wine encontramos una extensa base de

datos de programas testados bajo Wine y su desempeño [16].

En último lugar, disponemos de un recurso que prácticamente nunca nos va a fallar, y es el uso de una máquina virtual Windows. Cualquier software de los indicados que no disponga de compatibilidad de ningún tipo con Linux y que no funcione correctamente con Wine, va a poder ser ejecutado mediante una VM. De hecho, en el entorno estudiado hemos encontrado varios programas que para su ejecución requerían de una VM con Linux o una versión antigua de Windows. Dichas VMs podrían ser ejecutadas directamente sobre Linux, sin necesidad alguna de búsqueda de alternativas.

La propuesta que queremos presentar en este trabajo es precisamente dar la vuelta al modelo que solemos encontrar actualmente: en lugar de un entorno de trabajo Windows en el que para poder realizar acciones bajo Linux necesitemos una máquina virtual, proponer un entorno basado en Linux en el que solo para contadas ocasiones necesitemos utilizar una VM Windows, y no necesitemos de máquinas virtuales para poder trabajar bajo un sistema operativo Linux.

Además ya hemos indicado que la virtualización bajo Linux mediante KVM es bastante eficiente, y en ciertos casos puede poseer un rendimiento muy cercano al que proporcione el anfitrión sin ningún tipo de virtualización. Por tanto, podríamos disponer de un entorno Windows virtualizado, el cual nos proporcionaría una experiencia de uso muy similar a la que nos daría un equipo con Windows instalado de forma nativa.

Apartado 5.12. Tabla resumen de software

A continuación incluiremos una tabla a modo de resumen con el software indicado en los apartados anteriores:

Programa	Disponible en Linux	Comentarios
Software de propósito general		
Firefox	Sí	
Chrome	Sí	Existe Chromium como opción libre
Internet Explorer	No	
LibreOffice	Sí	
Microsoft Office	No	Posibilidad de utilizar LibreOffice u Office Online
Notepad++	No	Otros editores: NotepadQQ, Emacs, Vim
Wordpad	No	Otras opciones: Gedit, NotepadQQ

PDF-XChange Viewer	No	Alternativa: Evince
7-zip	No	Alternativas: p7zip, tar
VLC	Si	
GIMP	Sí	
Entornos de desarrollo		
Eclipse	Sí	
Netbeans	Sí	
IntelliJ IDEA	Sí	
Oracle JDeveloper Studio	Sí	
Code::Blocks	Sí	
GNAT Programming Studio	Sí	
Kojo	Sí	
Visual Studio	Parcial	Similar: Visual Studio Code y Azure DevOps
Entornos de ejecución		
Java JDK/JRE	Sí	Posibilidad de utilizar versión de Oracle u OpenJDK (libre)
JavaCC	Sí	
Free Pascal	Sí	
SWI-Prolog	Sí	
Camlwin	No	Anticuoado. Versión más actual OCaml disponible para Linux
WinGHCi Haskell	No	Alternativa: Haskell Platform
WinMIPS64	No	Alternativa: EduMIPS64
MASM	No	Diferente para Linux, posibilidad de usar as o nasm
Programación de hardware		
Xilinx ISE/Vivado Design Suite	Sí	
ModelSim	No	Funciona con Wine. Alternativa: entornos de Xilinx
Veribest VHDL simulator	No	Obsoleto, mejor utilizar uno de los dos anteriores
Otras utilidades de diseño y desarrollo		
Git	Sí	
XML Copy Editor	Sí	
HTML-Kit Tools	No	Funciona con Wine
StarUML	Sí	
ArgoUML	Sí	

Rational DOORS	Parcial	Solo funciona en Linux la parte servidor, no el cliente
yEd Graph Editor	Sí	
gvSIG Desktop	Sí	
Weka	Sí	
Microsoft Power BI Desktop	No	Existen otras opciones, por ejemplo Kibana
Herramientas de redes		
Putty	No	Opción libre: OpenSSH. Telnet, nc o ncat para conexiones Telnet
WinSCP	No	Opción libre: OpenSSH (utilidades scp o sftp)
Wireshark	Sí	
OPNet Modeler	Sí	Ahora conocido como Riverbed Modeler
Herramientas de seguridad		
OpenSSL	Sí	
X - Certificate and Key management	Sí	Posibilidad de utilizar OpenSSL directamente, o easy-rsa
PGP	Sí	Opción libre: GPG
Bases de datos y almacenamiento		
Oracle Database	Sí	
MySQL	Sí	Opción completamente libre: MariaDB
MySQL Workbench	Sí	
PostgreSQL	Sí	
XAMPP	Sí	
BaseX	Sí	
Microsoft Azure Storage Explorer	Sí	
Software de virtualización		
Oracle VM VirtualBox	Sí	
Vagrant	Sí	
Docker	Sí	
Software matemático		
Matlab	Sí	
Maple	Sí	
wxMaxima	Sí	
Derive	No	Anticuoado
Statgraphics	No	

Capítulo 6. Estimación de tiempos

En esta sección proporcionaremos un listado de tareas a realizar para llevar a cabo la implantación de la infraestructura, así como una estimación de tiempo para cada una de ellas. Podemos identificar las siguientes tareas:

- Preparativos iniciales (definición de nuevas redes, preparación de nuevos servidores, etc.): 2 semanas
- Instalación y configuración de FreeIPA: 1 semana
- Integración de FreeIPA con AD: 1 semana
- Instalación y configuración de Foreman: 1 semana
- Adecuación del almacenamiento a nueva infraestructura: 2 semanas
- Pruebas de funcionamiento y de alta disponibilidad: 2 semanas
- Empaquetado de software y creación de plantillas de provisión: 3 semanas
- Despliegue de un número reducido de equipos, y prueba piloto con usuarios: 4 semanas
- Despliegue del resto de puestos de trabajo: 2 semanas

A continuación incluimos un diagrama de Gantt con dicha estimación:

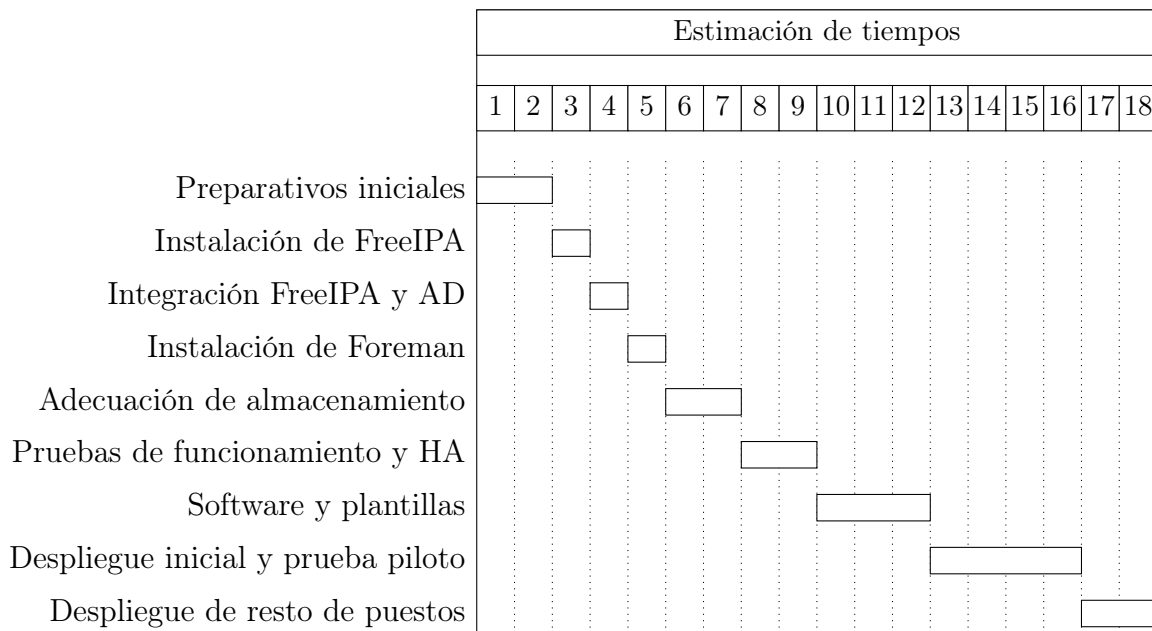


Figura 28: Diagrama de Gantt con estimación de tiempos

Capítulo 7. Otros usos

Como ya hemos indicado anteriormente, aunque el software descrito en este documento se puede utilizar para la gestión de puestos de trabajo con entornos de escritorio, éste está orientado principalmente para la gestión de servidores en un entorno de centro de datos. Esto quiere decir que el software está diseñado para gestionar grandes entornos, que puede que sean bastante heterogéneos.

La instalación de este software permitiría no solo la gestión de puestos de trabajo, sino que también podría emplearse para gestionar los servidores Linux que esté empleando el centro actualmente, o también de equipos de trabajo situados fuera del centro principal de trabajo, como por ejemplo diferentes laboratorios.

El hecho de poder asignar usuarios y equipos a diferentes grupos nos permite gestionar múltiples entornos de forma sencilla, y mantener además aislados los permisos de acceso entre ellos. Por ejemplo, Foreman nos permite asignar equipos a diferentes organizaciones y ubicaciones, de forma que los equipos de cada sala puedan tener ciertas particularidades en su configuración. FreeIPA permite también asociar automáticamente los equipos a diferentes grupos en función de la ubicación proporcionada por Foreman, y asignar diferentes unidades de red para el almacenamiento según dicha ubicación.

Además, tanto con FreeIPA como Foreman es posible escalar la infraestructura, de forma que se puedan delegar las peticiones en múltiples servidores secundarios, en lugar de realizar todas contra el servidor principal. A este respecto FreeIPA nos permite crear múltiples réplicas del servidor principal, y Foreman dispone de *Smart Proxies*, que pueden gestionar los servicios de DNS, DHCP, PXE, etcétera.

Hay que indicar también que, aunque no se lleve a cabo una implantación de este software para la gestión de un centro de trabajo, podría también emplearse en otras áreas o departamentos. Foreman, por ejemplo, sería bastante útil para reaprovisionar desde cero equipos de un laboratorio para diferentes etapas de trabajo, o para mantener una paquetería y configuración homogénea en un grupo de equipos Linux.

Igualmente, FreeIPA podría resultar de gran utilidad para gestionar las identidades en los servidores del centro de datos. De esta forma, los accesos como administrador hacia los servidores se realizarían con usuarios nominales en lugar de un usuario genérico como *root*, lo que proporciona un mayor control de los accesos a dichos servidores y un mayor nivel de seguridad en el sistema.

Capítulo 8. Impactos sociales y ambientales

La propuesta de esta infraestructura está destinada a promover ante los estudiantes el uso de Linux en general. La intención es que los estudiantes obtengan una buena experiencia y una mayor confianza en el manejo de Linux. De esta forma, nuevas promociones de informáticos irán impulsando a su vez la utilización de Linux tanto en su ámbito profesional como personal.

Aunque sea en una pequeña medida, esto debería aumentar el uso de Linux como sistema operativo, así como de nuevos colaboradores en su desarrollo y en el de otros proyectos de software libre. De esta forma se mejorará la calidad de este software, el cual poco a poco irá incrementando su cuota de mercado en los hogares y aumentando su implantación en empresas, no solo en el mundo servidor sino también de escritorio. Esto irá desplazando poco a poco el uso de programas propietarios, cediendo lugar a software de código abierto.

Esta transformación de hecho ya está ocurriendo: grandes empresas ven el software libre como el futuro y apoyan este tipo de proyectos. Como ejemplo, Microsoft liberó recientemente unas 60.000 patentes para proteger proyectos de código abierto y demostrar así su compromiso con el software libre [17].

Acerca de los impactos medioambientales, hay datos que ya apuntan a los centros de datos de todo el mundo como los causantes del 2% de emisiones de gases de efecto invernadero [18]. Esto es debido a la gran cantidad de energía que requieren estos centros de datos, tanto en consumo directo de servidores como de refrigeración. Con respecto al primero, la eficiencia del software juega un importante papel en los consumos de CPU, que se puede traducir directamente en consumo energético.

Dado que Linux es un sistema operativo más ligero que Windows, su eficiencia es mayor, y por tanto con su uso podemos reducir los consumos energéticos en los centros de datos. Promoviendo por tanto su uso conseguiremos que aumente su presencia, tanto en los servidores propios de cada empresa como para proyectos desarrollados en nubes públicas de gigantes tecnológicos como Amazon, Google o Microsoft, lo que a su vez mejoraría el consumo energético por parte de estas empresas.

También, al ser más liviano, Linux nos da la capacidad de instalarlo en ordenadores con menos recursos. Esto permite recuperar equipos antiguos que hayan sido dejados de lado al no soportar versiones más recientes de otros sistemas operativos, con lo que podrían ser útiles de nuevo. Con esto podríamos aprovechar dichos equipos para otros fines en lugar de tener que adquirir nuevo hardware, lo que nos ayudaría a reducir la cantidad de residuos electrónicos generados en los hogares.

Capítulo 9. Responsabilidad ética y profesional

Ya hemos indicado que el uso de software privativo tiene ciertas consecuencias. En primer lugar, que el control de gran parte del software que utilizamos en el día a día está en manos de una única empresa, o de un grupo muy reducido de ellas. Esto da lugar a que en ocasiones nos veamos obligados a tener que aceptar actualizaciones no deseadas, o tener que adquirir nuevas licencias porque nuestra versión actual de software va a dejar de recibir parches de seguridad.

Este tipo de prácticas forman parte de la llamada obsolescencia programada, por la cual las empresas fuerzan a que software o dispositivos queden obsoletos o se vuelvan inútiles pasado un tiempo determinado, reduciendo así su tiempo de vida útil. Estas prácticas son muy cuestionables a nivel ético, y difícilmente evitables por parte de los usuarios. Sin embargo, el uso de software libre puede ayudarnos a mitigar el impacto de la obsolescencia programada [19].

Los usuarios de a pie en muchas ocasiones desconocen alternativas libres, ya que por lo general los dispositivos que adquieren suelen tener software privativo preinstalado, que es el que finalmente utiliza el usuario. Para que pueda ocurrir un cambio de paradigma en el que el software libre sea la alternativa por defecto, deben de existir grupos de interés que potencien estas alternativas libres, creándolas, mejorándolas, y promoviendo su uso.

Es por tanto que como informáticos, nuestro deber es el de tratar de que dicho cambio se produzca. Por esto mismo, debemos ser los principales concededores de dichas alternativas, con lo que tendría que ser casi de carácter obligatorio una educación a este respecto durante las enseñanzas universitarias. De esta forma podremos tratar de esquivar las prácticas abusivas que llevan a cabo ciertas empresas propietarias de software de uso popular.

Adicionalmente, el uso de software libre en el ámbito laboral proporciona múltiples ventajas, ya descritas en este documento: mayor libertad de uso de software, mejor rendimiento y menor coste. Nuestra responsabilidad profesional como informáticos deberá ser la de abogar por alternativas de software que beneficien en mayor medida a la empresa, y muchas veces encontraremos dichas alternativas en programas libres y de código abierto.

Por este motivo, de nuevo recomendamos el educar a los futuros informáticos acerca de Linux y el software libre. Si bien en ocasiones puede que no resulte la mejor elección en ciertos entornos, el informático por lo menos deberá conocer dichas alternativas para poder valorar todas las opciones disponibles, y seleccionar la más adecuada en cada caso.

Capítulo 10. Conclusiones

Queremos reiterar nuevamente que llevar a cabo una migración de un centro de trabajo universitario de Windows a Linux puede reportar múltiples beneficios. El principal que queremos obtener es lograr que los alumnos se familiaricen y se sientan cómodos con Linux como entorno de trabajo principal, de forma que lo puedan llegar a tomar como opción principal de cara a futuros proyectos.

Además, el uso de software libre (aunque requiera una inversión inicial para la migración) permitiría ahorrar bastantes costes en licencias de software. Podemos encontrar casos de ahorro como el que consiguió la ciudad de Munich (de unos 11 millones de euros) [20], o como ejemplo más cercano (y relativo al ámbito universitario), el de la universidad de Murcia, que han logrado un ahorros anual de aproximadamente 100.000 euros [21].

Se ha podido observar que, aunque una migración a Linux pueda requerir recursos y tiempo, con herramientas adecuadas es posible llevarla a cabo de forma sencilla, y sin renunciar a sistemas de gestión profesionales como los existentes en arquitecturas Windows (por ejemplo, el sistema de gestión de usuarios).

También se han podido encontrar soluciones para seguir ejecutando software típico que podríamos encontrar en un centro universitario, y se han observado diferentes alternativas de software en caso que no fuera posible ejecutar el programa original. Entendemos pues, que se ha podido comprobar la viabilidad de utilizar Linux como entorno de escritorio para el trabajo del día a día de alumnos universitarios.

Esperamos por tanto que este documento pueda llegar a incentivar a sus lectores para se puedan plantear un cambio de tecnologías, y que promuevan a su vez la posibilidad de migración a Linux en sus centros de trabajo, ya sean universitarios o empresariales.

11. Bibliografía

Referencias

- [1] TOP500.org, Operating system Family / Linux,
<https://www.top500.org/statistics/details/osfam/1>
- [2] The Linux Foundation, The Top 10 Developers and Companies Contributing to the Linux Kernel in 2015-2016
<https://www.linuxfoundation.org/blog/2016/08/the-top-10-developers-and-companies-contributing-to-the-linux-kernel-in-2015-2016/>
- [3] Free Software Foundation, Proprietary Back Doors
<https://www.gnu.org/proprietary/proprietary-back-doors.html>
- [4] The CentOS Project, About CentOS
<https://www.centos.org/about/>
- [5] Red Hat Inc., Red Hat Enterprise Linux operating system
<https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux>
- [6] Scientific Linux @ CERN: Next Version
<http://linux.web.cern.ch/linux/nextversion.shtml>
- [7] Fedora Project, Fedora's Mission and Foundations
<https://docs.fedoraproject.org/en-US/project/>
- [8] FreeIPA - About FreeIPA
<https://www.freeipa.org/page/FreeIPA:About>
- [9] Foreman - Introduction
<https://theforeman.org/introduction.html>
- [10] Spacewalk: Free & Open Source Systems Management
<https://spacewalkproject.github.io/index.html#about>
- [11] FreeIPA - Deployment Recommendations
https://www.freeipa.org/page/Deployment_Recommendations
- [12] Wikipedia - Hypervisor, Classification
<https://en.wikipedia.org/wiki/Hypervisor#Classification>

-
- [13] Red Hat - Linux Domain Identity, Authentication, and Policy Guide
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/pdf/linux_domain_identity_authentication_and_policy_guide/Red_Hat_Enterprise_Linux-7-Linux_Domain_Identity_Authentication_and_Policy_Guide-en-US.pdf
- [14] Foreman - Manual
<https://theforeman.org/manuals/1.19/index.html>
- [15] Michael Larabel, Photonix, Ubuntu 15.10: KVM vs. Xen vs. VirtualBox Virtualization Performance
<https://www.phoronix.com/scan.php?page=article&item=ubuntu-1510-virt>
- [16] WineHQ, Wine Application Database
<https://appdb.winehq.org/>
- [17] Steven J. Vaughan-Nichols, ZDNet, Microsoft open-sources its patent portfolio
<https://www.zdnet.com/article/microsoft-open-sources-its-entire-patent-portfolio/>
- [18] Adam Vaughan, The Guardian, How viral cat videos are warming the planet
<https://www.theguardian.com/environment/2015/sep/25/server-data-centre-emissions-air-travel-web-google-facebook-greenhouse-gas>
- [19] Bertel King, Jr., MakeUseOf, Defeat Planned Obsolescence with Linux and Open Source Software
<https://www.makeuseof.com/tag/protect-planned-obsolescence-using-linux-open-source-software/>
- [20] Loek Essers, ITWorld, Switching to Linux saves Munich over 11 million
<https://www.itworld.com/article/2716115/operating-systems/switching-to-linux-saves-munich-over--11-million.html>
- [21] J.Pomeyrol, MuyLinux, La Universidad de Murcia, en proceso de migración a Linux
<https://www.muylinux.com/2014/06/04/universidad-de-murcia-migracion-linux/>

12. Anexos

12.1. Anexo 1 - Acrónimos

- AD: Active Directory
- CA: Certificate Authority
- CERN: Conseil Européen pour la Recherche Nucléaire
- DHCP: Dynamic Host Configuration Protocol
- DNS: Domain Name System
- FPGA: Field Programmable Gate Array
- FQDN: Fully Qualified Domain Name
- FTP: File Transfer Protocol
- GNU: GNU is Not Unix
- GPG: GNU Privacy Guard
- HBAC: Host-based Access Control
- HTML: Hypertext Markup Language
- HTTP: Hypertext Transfer Protocol
- HTTPS: HTTP Secure
- IPAM: IP Address Management
- JDK: Java Development Kit
- JRE: Java Runtime Environment
- KVM: Kernel Virtual Machine
- LDAP: Lightweight Directory Access Protocol
- MAC: Media Access Control
- NAS: Network Attached Storage
- NAT: Network Address Translation
- NFS: Network File System

- NTP: Network Time Protocol
- OS: Operating System
- OTP: One-Time Password
- PDF: Portable Document Format
- PGP: Pretty Good Privacy
- POSIX: Portable Operating System Interface
- PXE: Preboot eXecution Environment
- RDP: Remote Desktop Protocol
- RPM: Red Hat Package Manager / RPM Package Manager
- RSA: Rivest–Shamir–Adleman
- SCP: Secure Copy Protocol
- SFTP: SSH File Transfer Protocol
- SMB: Server Message Block
- SQL: Structured Query Language
- SSH: Secure Shell
- SSL: Secure Sockets Layer
- TFTP: Trivial File Transfer Protocol
- UML: Unified Modeling Language
- URL: Uniform Resource Locator
- VHDL: VHSIC Hardware Description Language
- VNC: Virtual Network Computing
- VM: Virtual Machine
- XML: eXtensible Markup Language

12.2. Anexo 2 - URLs con compatibilidad de software

En el siguiente listado incluiremos el software discutido en el apartado 5, junto con una serie de URLs en donde podremos observar su compatibilidad (o falta de ella) con Linux. Generalmente trataremos de indicar la URL con los requisitos del sistema, o en los casos en que no se disponga de dicha página redirigiremos a su página de descarga (donde se observarán las descargas para Linux) o del proyecto en sí:

- Firefox:
<https://www.mozilla.org/en-US/firefox/63.0.3/system-requirements/>
- Chrome:
<https://support.google.com/chrome/a/answer/7100626>
- Chromium:
<https://www.chromium.org/administrators/linux-quick-start>
- Internet Explorer:
<https://support.microsoft.com/en-ca/help/11531/internet-explorer-system-requirements>
- LibreOffice:
<https://www.libreoffice.org/get-help/system-requirements/>
- Microsoft Office:
<https://support.microsoft.com/es-es/help/4026853/office-office-2016-system-requirements>
- Microsoft Office Online:
<https://support.office.com/en-us/article/which-browsers-work-with-office-online-ad1303e0-a318-47aa-b409-d3a5eb44e452>
- Notepad++:
<https://notepad-plus-plus.org/>
- NotepadQQ:
<https://notepadqq.com/wp/download/>
- PDF-XChange:
<https://pdf-xchange.eu/pdf-xchange-pro/system-requirements.htm>
- Adobe Reader:
<https://helpx.adobe.com/reader/system-requirements-old-versions.html>
- 7-zip:
<https://www.7-zip.org/download.html>
- p7zip:
<http://p7zip.sourceforge.net/>

- VLC:
<https://www.videolan.org/vlc/features.html>
- GIMP:
https://www.gimp.org/downloads/install_help.html#requirements
- Eclipse:
<https://www.eclipse.org/downloads/packages/>
- Netbeans:
https://netbeans.org/community/releases/82/relnotes.html#system_requirements
- IntelliJ IDEA:
<https://www.jetbrains.com/help/idea/install-and-set-up-product.html>
- Oracle JDeveloper Studio:
<https://docs.oracle.com/en/middleware/lifecycle/12.2.1.3/sysrs/unix-operating-system-requirements.html>
- Code::Blocks:
<http://www.codeblocks.org/downloads/26#linux>
- GNAT Programming Studio:
<https://www.adacore.com/download>
- Kojo:
<http://www.kogics.net/kojo-download#unix>
- Microsoft Visual Studio:
<https://docs.microsoft.com/es-es/visualstudio/productinfo/vs2017-system-requirements-vs>
- Microsoft Visual Studio Code:
<https://code.visualstudio.com/docs/supporting/requirements>
- Microsoft Azure DevOps:
<https://azure.microsoft.com/es-es/services/devops/>
- Java:
<https://java.com/en/download/help/sysreq.xml>
- OpenJDK:
<https://openjdk.java.net/>
- JavaCC:
<https://javacc.org/>
- Free Pascal:
<https://www.freepascal.org/docs-html/user/usersu2.html>

- SWI-Prolog:
<http://www.swi-prolog.org/download/stable>
- CamlWin:
<https://caml.inria.fr/>
- OCaml:
<https://ocaml.org/docs/install.html>
- Haskell Platform:
<https://www.haskell.org/platform/>
- WinMIPS64:
<http://indigo.ie/~mscott/>
- EduMIPS64:
<https://www.edumips.org/>
- Microsoft Macro Assembler:
<https://www.microsoft.com/en-us/download/details.aspx?id=12654>
- Netwide Assembler:
<https://www.nasm.us/>
- Xilinx ISE Design Suite:
https://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/irn.pdf
- Xilinx Vivado Design Suite:
https://www.xilinx.com/products/design-tools/vivado/vivado-webpack.html#os_support
- ModelSim:
http://s3.mentor.com/public_documents/datasheet/products/fv/modelsim_pe_ds.pdf
- Git:
<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
- XML Copy Editor:
<https://sourceforge.net/projects/xml-copy-editor/files/xmlcopyeditor-linux/>
- HTML-Kit Tools:
<http://www.htmlkit.com/blog/html-kit-292-on-linux/>
- StarUML:
<http://staruml.io/download>

- ArgoUML:
<http://argouml-stats.tigris.org/documentation/quickguide-0.34/ch02.html>
- Rational DOORS:
https://www.ibm.com/support/knowledgecenter/en/SSYQBZ_9.6.1/com.ibm.doors.install.doc/topics/t_installingclient.html
- yEd Graph Editor:
<https://www.yworks.com/downloads#yEd>
- gvSIG Desktop:
<http://downloads.gvsig.org/download/web.save/projects/gvsig-desktop/official/gvsig-1.1/system-requirements.html>
- Weka:
<https://www.cs.waikato.ac.nz/~ml/weka/requirements.html>
- Microsoft Power BI Desktop:
<https://docs.microsoft.com/es-es/power-bi/desktop-get-the-desktop>
- Kibana:
<https://www.elastic.co/products/kibana>
- Putty:
<https://www.putty.org/>
- WinSCP:
<https://winscp.net/eng/docs/requirements>
- Wireshark:
https://www.wireshark.org/docs/wsug_html_chunked/ChIntroPlatforms.html
- Riverbed Modeler:
<https://support.riverbed.com/content/support/software/steelcentral-npm/modeler-index.html>
- OpenSSL:
https://wiki.openssl.org/index.php/Compilation_and_Installation
- X - Certificate and Key management:
<https://www.hohnstaedt.de/xca/index.php/download>
- Easy-RSA:
<https://github.com/OpenVPN/easy-rsa>
- OpenPGP:
<https://www.openpgp.org/software/>

- Oracle Database:
<https://docs.oracle.com/en/database/oracle/oracle-database/18/ladbi/index.html>
- MySQL:
<https://www.mysql.com/support/supportedplatforms/database.html>
- MySQL Workbench:
<https://www.mysql.com/support/supportedplatforms/workbench.html>
- MariaDB:
<https://mariadb.org/download/>
- PostgreSQL:
<https://www.postgresql.org/docs/11/supported-platforms.html>
- XAMPP:
<https://www.apachefriends.org/es/download.html>
- BaseX:
<http://docs.basex.org/wiki/Startup#Requirements>
- Microsoft Azure Storage Explorer:
<https://azure.microsoft.com/es-es/features/storage-explorer/>
- Oracle VM VirtualBox:
https://www.virtualbox.org/wiki/End-user_documentation
- Vagrant:
<https://www.vagrantup.com/docs/installation/>
- Docker:
<https://docs.docker.com/install/>
- Matlab:
<https://www.mathworks.com/support/sysreq.html>
- Derive:
<https://www.chartwellyorke.com/derive.html>
- Maple:
https://www.maplesoft.com/products/system_requirements.aspx
- Statgraphics:
<http://www.statgraphics.com/download18>
- wxMaxima:
<https://wxmaxima-developers.github.io/wxmaxima/download.html>

- Wine:
<https://www.winehq.org/>
- Mono:
<https://www.mono-project.com/>