

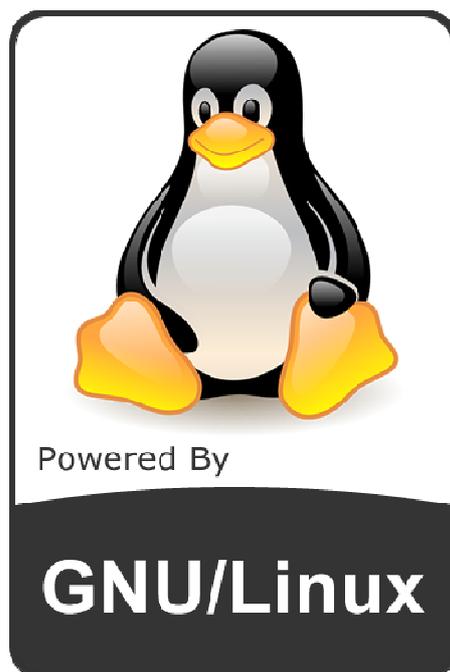
Implementación de GNU/Linux para una empresa de desarrollo de software
David Manzano Fiestas

*Implementación de GNU/Linux para una empresa de desarrollo
de software.*

Trabajo Fin de Carrera - David Manzano Fiestas

Tutor UOC - Pere Manzanos González

Tutor TFC – Joaquín López Sanchez-Montañez



SOBRE ESTE DOCUMENTO

Implementación de GNU/Linux para una empresa de desarrollo de software	Título del documento
David Manzano Fiestas	Responsable
DavidManzano_GNULinux_TFC	Nombre del fichero
80	Número de páginas

REVISIÓN

Fecha	Autor	Descripción	Versión
05-11-2013	David Manzano Fiestas		0.1
23-12-2013	David Manzano Fiestas		0.2
10-1-2014	David Manzano Fiestas		Final

Este documento se ha escrito utilizando para su edición y almacenamiento estándares abiertos específicos para documentos ofimáticos (**Open Document Format**, publicado el 30 de noviembre de 2006 por las organizaciones ISO/IEC como estándar ISO/IEC 26300:2006 Open Document Format for Office Applications (OpenDocument) v1.0.).

Las fuentes utilizadas en este documento están extraídas de la Open Font Library (<http://openfontlibrary.org/>) y cumplen con la licencia **Creative Commons** (inspirada en la licencia GPL) para la distribución y el uso de contenidos digitales abiertos.

Para el texto se ha utilizado la fuente Puritan 2.0: <http://openfontlibrary.org/files/benweiner/259>

Para títulos, tablas, ilustraciones, etc. se ha utilizado la fuente Jura: http://openfontlibrary.org/files/Daniel_J/98

Las imágenes utilizadas en este documento también cumplen con la licencia Creative Commons.

Índice

[1. Introducción](#)

[1.1 Descripción de GNU/Linux](#)

[1.2 Open Source y sus ventajas](#)

[1.3 Necesidades en la empresa](#)

[2. Descripción de la arquitectura a implementar](#)

[2.1 Infraestructura hardware](#)

[2.2 Infraestructura software](#)

[3. Soluciones en Software Libre adoptadas](#)

[3.1 Sistemas operativos](#)

[3.1.1 Debian 7](#)

[3.2 Virtualización](#)

[3.2.1 Contenedores con OpenVZ](#)

[3.2.2 Virtualización completa con KVM](#)

[3.2.3 ProxmoxVE](#)

[3.3 Seguridad](#)

[3.3.1 Pfsense](#)

[3.3.2 Copias de seguridad](#)

[3.3.3 Monitorización con Nagios](#)

[3.4 Control de usuarios](#)

[3.4.1 OpenLDAP](#)

[3.5 Comunicaciones](#)

[3.5.1 Gestión del correo: Zimbra](#)

[3.5.2 Mensajería: Openfire](#)

[3.5.3 VPN](#)

[3.6 Bases de datos](#)

[3.6.1 MySQL](#)

[3.7 Publicación](#)

[3.7.1 Servidor web: Apache](#)

[3.7.2 Servidor de aplicaciones: Tomcat](#)

[3.8 Gestión de proyectos](#)

[3.8.1 Redmine](#)

[3.8.4 Netbeans y Eclipse](#)

- [3.8.5 Subversion](#)
- [3.9 Otros servicios](#)
 - [3.9.1 DNS con Bind9](#)
 - [3.9.2 Sincronización horaria con NTP](#)
 - [3.9.3 Libreoffice](#)
- [4. Descripción de la implementación de la simulación](#)
- [5. Conclusiones y resultados](#)
 - [5.1 Evaluación global del funcionamiento](#)
 - [5.2 Ventajas del uso de GNU/Linux](#)
 - [5.2 Posibles problemas de migración desde software propietario](#)
 - [5.3 Sistemas alternativos](#)
- [6. Anexos](#)
 - [6.1 Documentos y publicaciones de apoyo](#)
 - [6.2 Direcciones web relacionadas](#)

1. Introducción

A lo largo del siguiente trabajo se va a tratar de demostrar que haciendo uso de tecnologías Libres, y basándonos en sistemas operativos abiertos, es posible mantener un nivel alto de trabajo y desarrollo cuando hablamos de una empresa que se dedica a implementar y realizar desarrollos en las tecnologías que actualmente se utilizan para la implementación de aplicaciones.

Para poder llevar a cabo esta labor montaremos un laboratorio de desarrollo que nos va a permitir entender el funcionamiento y la implementación tanto de GNU/Linux como del software que se basa en él dentro de la infraestructura de la empresa, y veremos como uniendo todas las partes se consigue una solución global y completa que ofrece todos los servicios o más incluso que los que se obtendría si utilizamos software privativo con el mismo fin.

Durante el transcurso de las pruebas e implementaciones realizaremos una labor de investigación sobre las distintas tecnologías disponibles, valorando cuales se adaptarán mejor al propósito que queremos obtener, escogiendo aquellas que consideremos las más adecuadas a nuestras metas y adaptandolas además al hardware del que disponemos para realizar la simulación.

Tras finalizar el proyecto, dispondremos de una simulación completa de una red empresarial, que a mayor escala podría implementarse como infraestructura base para una pequeña empresa de desarrollo, y que dispondrá de todas las herramientas software necesarias para la correcta ejecución de su actividad.

Nuestra empresa se dedicará fundamentalmente a la elaboración de software basado en tecnologías de software libre, y en la medida de lo posible intentará liberar el código fuente producto de dichos desarrollo colaborando así activamente en la difusión y mejora de la comunidad de software Libre.

Aunque todo el software que vamos a utilizar a lo largo de la ejecución del proyecto está basado en software libre, tendremos que establecer una serie de salvedades, como son controladores y software propietarios que pueden encontrarse embebidos en dispositivos y que muchas veces es inevitable utilizar, como puede ser el software de gestión de un Router o de un Switch, que puede contener software propietario de la marca concreta que lo ha fabricado. Del mismo modo habrá que incorporar a nuestro proyecto software proveniente de repositorios de nuestras distribuciones que aunque no es considerado como Software Libre, si que es indispensable para el correcto funcionamiento de los distintos dispositivos, como son los drivers de tarjetas de red, tarjetas de video, etc.

En cualquier caso lo que se va a evitar es incorporar software que incluya un licenciamiento de pago, es decir que para poder utilizarlo debemos pagar una licencia, y software que no vaya embebido en dispositivos pero cuyas fuentes no sean abiertas .

1.1 Descripción de GNU/linux

En el año 1985, se fundó la denominada Free Software Foundation (FSF), una organización que promueve el derecho del usuario a usar, estudiar, copiar, modifica y redistribuir programas informáticos. La FSF defiende el desarrollo y uso del software libre.

Cuando se habla de la libertad del software, nos estamos refiriendo fundamentalmente a la libertad en varias vertientes:

- Libertad para ejecutar el software con cualquier propósito.
- Libertad para estudiar cómo funciona el programa y adaptarlo a nuestras necesidades. EL tener acceso al código fuente es un requisito para esto.
- Libertad para redistribuir las copias de manera que podamos ayudar a los demás.
- Libertad para mejorar el software, y liberar ese software mejorado al público, de modo que la comunidad entera se pueda beneficiar.

El proyecto GNU se lanzó en 1983 por Richard Stallman para desarrollar un sistema operativo completo basado en Unix, a esto se le denominó el proyecto GNU. El kernel de GNU está aún sin completar, por lo que en este se implementó el kernel de Linux. La combinación de ambos dio lugar al sistema operativo GNU/Linux. La persona que comenzó a dar un sentido completo al proyecto fué Linus Benedict Torvalds que comenzó el desarrollo de Linux, el kernel del sistema operativo y actualmente trabaja como coordinador de dicho proyecto.

Aunque originalmente UNIX era propietario por parte de AT&T, en 1987 el profesor Andrew S. Tanenbaum decidió crear su propio clon que empleaba con fines pedagógicos, MINIX, y fué basándose en éste con el que Linus Torvalds decidió escribir su propio núcleo del sistema operativo.

Torvalds, tras instalar MINIX en su propio ordenador, empezó a reescribir el núcleo de forma que pronto la funcionalidad de Linux sobrepasó la de MINIX, de este modo la primera versión de kernel de Linux fué publicada en Septiembre de 1991, con la segunda versión llegando poco después en el mes de Octubre.

En estos momentos el proyecto GNU, que tenía bastante avanzado el desarrollo de su sistema operativo libre, se encontraba haciendo uso de Hurd, un núcleo que no era tan estable como era Linux, de este modo el proyecto GNU adoptó el kernel Linux, para poder seguir avanzando en el proyecto mientras terminaban Hurd, de esta manera nació lo que ahora conocemos como GNU/Linux.

En la actualidad existen multitud de distribuciones para todo tipo de usuarios e instituciones, muchas de las cuales se han llegado a convertir en grandísimos productos comerciales como son Suse o RedHat, y muchas otras que se encuentran personalizadas y al servicio de diverso hardware o dispositivos.

Otras muchas distribuciones han surgido como modificaciones de otras, puesto que esto representa una de las filosofías del software libre, y es que cualquier usuario puede coger un software, modificarlo y publicarlo para la comunidad. Si queremos obtener una lista completa de las distribuciones existentes podemos visitar:

<https://lwn.net/Distributions/>

Y el esquema que enlazamos a continuación nos podemos hacer una idea de la diversidad de sistemas que nos podemos encontrar enfrentados a una línea temporal de publicación y en el que se pueden apreciar los distintas versiones de sistemas que han surgido a partir de una distribución concreta:

<http://robertbriones.files.wordpress.com/2007/07/linea-tiempo-linux.png>

1.2 Open Source y sus ventajas

Open-source es un software gratuito en cuanto a uso, distribución y modificación, o si tiene algún coste de venta, éste es una mínima fracción del precio de un software propietario. Al ser el código accesible a todas las personas, este software es más seguro, ya que cualquiera puede corregir errores que vayan apareciendo, sin necesidad de tener que estar esperando continuas actualizaciones por parte de los propietarios del software. Además no tiene los problemas de formatos incompatibles que existen en software propietario. Los usuarios de diferentes empresas pueden coger una parte del software Open-source y modificarlo para adaptarlo a sus necesidades. Dado que el código es abierto, es simplemente una cuestión de modificarlo para añadir la funcionalidad que deseen.

Como Open-source no depende de la compañía o personas que lo crearon originalmente, aunque esta compañía desapareciese el código seguiría existiendo. Además estas compañías no son las que dictan a las empresas qué pueden hacer o no con el software que adquirieron, algo que sí sucede con el software propietario.

Las compañías que lo usen no necesitan estar pensando en las licencias que necesitan, ni tienen que preocuparse de medidas antipiratería como son los números de serie o los códigos de activación.

Todo esto repercute directamente en el desembolso que debe hacer la empresa, porque entre el precio de adquisición del software mismo, el precio desorbitado que tienen los programas antivirus, los cargos extra de mantenimiento y actualizaciones varias, sumado al coste añadido que se tiene, en cuanto a programas y demás, por poseer un software propietario, se va una parte importante de los ingresos de la empresa, cosa fácilmente solventable mediante la adquisición de un software Open-source.

1.3 Necesidades en la empresa

La empresa está realizando una migración casi forzosa hacia la utilización de GNU/Linux, la causa de esta migración cada vez más evidente está en una razones de peso como son, efectividad de costes, confiabilidad y seguridad.

Hace unas décadas, cuando Linux aún se estaba ganando la confianza de los usuarios en el escritorio de casa y en los pequeños negocios, algo ocurrió que terminó con la implantación del mismo en compañías de nivel multinacional.

Si tomamos como ejemplo Red Hat Linux, que comenzó siendo una distribución de escritorio, llegó un momento en que se produjo una división de su desarrollo dando lugar al Red Hat Enterprise Linux, este cambio de mentalidad, hizo que Red Hat, una compañía que hasta el momento solo se preocupaba de clientes de escritorio empezase a centrar su negocio en empresas, y se centraron en los aspectos más importantes del negocio de esas empresas, eso no solo hizo que dejase de ser una distribución para “geeks”, sino que se convirtiera en una multinacional como es hoy siendo uno de los mayores contribuidores al Kernel de Linux.

Pero no solo Red Hat ha evolucionado en dicho sentido, Canonical también ha evolucionado su sistema operativo principal, orientándolo hacia un sistema más profesional, como se puede observar de los proyectos Ubuntu Server o su reciente alianza con OpenStack para ofrecer Ubuntu Cloud.

Toda las empresas, tienen una cierta variedad de necesidades a cubrir mediante su infraestructura informática, estas necesidades pueden ser de muy diversa índole, pero si hay algo que todas ellas tienen en común son una serie de requisitos:

- El primero y fundamental, es el ahorro de costes. Está sobradamente demostrado que la utilización de Software Libre es mucho más efectiva en cuanto a coste que las soluciones propietarias, además los posibles cargos asociados a la implementación o soporte del mismo van íntimamente relacionados con la generación de empleo como puede ser en el pago de algún tipo de suscripción de soporte para una empresa especializada en software libre, mucho más rentable que pagar por licencias de uso que la mayoría de las veces no incluyen ningún tipo de soporte adicional que suele ser necesario contratar aparte.
- La seguridad es una de las principales preocupaciones de la empresa, y aquí Linux se ha mostrado como un alumno aventajado cuando tratamos de compararlo con sistemas windows, los cuales se suelen ver mucho más afectados por software malintencionado y exploits de sus vulnerabilidades, además de tener que incurrir en gastos adicionales de software antivirus, etc.
- La versatilidad es posiblemente otro de los requisitos necesarios cuando se requiere gestionar una empresa completamente mediante Software Libre. Para ello GNU/Linux pone a nuestra disposición una biblioteca de recursos en sus repositorios que podríamos considerar infinita. Hoy en día es difícil no encontrar alguna utilidad o herramienta que no esté disponible bajo entornos de software libre,

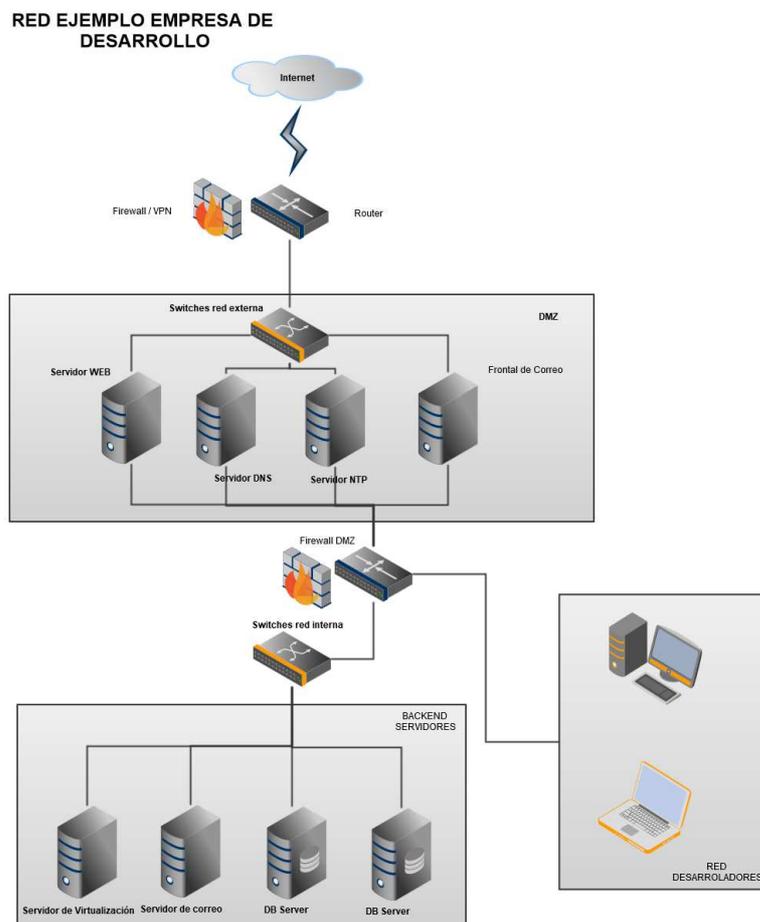
exceptuando claro las herramientas propietarias específicas, pero incluso en estos casos siempre disponemos de alguna alternativa que mediante un proceso de adaptación o entrenamiento nos garantizan unos resultados.

Contando con estas premisas vamos realizar una simulación completa de lo que sería un entorno de empresarial cuya infraestructura informática esté basada completamente en GNU/Linux, y las herramientas de Software Libre que se basan en el.

2. Descripción de la arquitectura a implementar

Para realizar una simulación de toda la infraestructura que una empresa debe manejar si desea implementar todos sus sistemas en software libre, vamos a describir como sería la infraestructura de dicha empresa y cómo adaptamos la misma a nuestro laboratorio para recrearla de la forma más eficiente posible.

El esquema organizativo general de la arquitectura de una empresa de desarrollo de software podría ser la siguiente:



En esta arquitectura podemos distinguir 3 partes diferenciadas.

Primero tenemos la parte DMZ en la que se alojarían todos los servicios que quedan publicados al exterior de la empresa. Es una zona que necesita de un gran control y monitorización puesto que es la más expuesta y vulnerable a ataques desde el exterior.

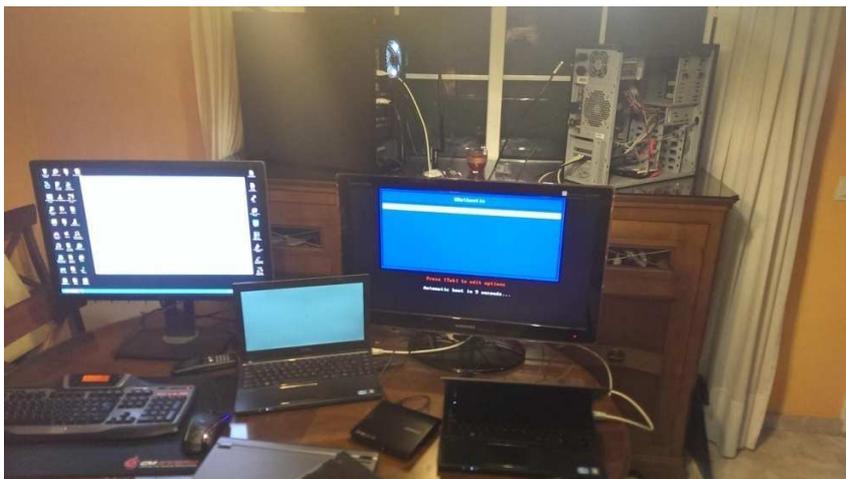
Por otra parte nos encontramos con el segundo Firewall que monta el sistema Dual Homed DMZ que separa toda la parte interna de la empresa de la parte externa y conectado a este nos encontramos subdividido en dos redes separadas el conglomerado de servidores que sirven de Backbone a la empresa y la red a la que se conectan todos los usuarios para desarrollar su labor diaria.

A grandes rasgos y siguiendo este esquema podríamos generar una infraestructura de red que dé soporte a todo el desarrollo y trabajo de nuestra empresa ficticia de desarrollo. La capacidad de albergar distintas máquinas en una u otra red siempre dependerá del hardware adquirido y la inversión realizada en el mismo, pero simplemente siguiendo este esquema de distribución podemos garantizar una infraestructura que nos permita desarrollar nuestra actividad de desarrollo con total normalidad.

Para conseguir emular dicha infraestructura se va a simular mediante una serie de máquinas físicas que mediante virtualización simularán ser máquinas hardware independientes y que interconectadas entre sí conformarán la red global.

En los siguientes apartados vamos a describir la infraestructura hardware con la que contamos en nuestro laboratorio de pruebas y describiremos cómo vamos a conformar dicha simulación.

2.1 Infraestructura hardware



Para poder simular toda la infraestructura de cuenta con el siguiente hardware:

➤ Ordenador Backend:

Intel Core i7 2600k, 8GB RAM, 1Tb HDD

Este ordenador mediante virtualización hará las veces de backend de servidores, es decir en el vamos a desplegar todas aquellas máquinas que corresponden al CPD de nuestra empresa de desarrollo y en la cual vamos a alojar todos los servicios necesarios para el correcto funcionamiento de la empresa. El software elegido para llevar a cabo dicha virtualización es VirtualBOX un software liberado bajo licencia GPL.

➤ Ordenador Firewall:

AMD Athlon 64, 2GB RAM, 250GB HDD

En el vamos a virtualizar un PFSense con cuatro interfaces de red. Una de las interfaces irá conectada a la salida a internet y actuará como WAN. Las otras tres interfaces mediante simularán estar conectadas una a cada una de las REDES descritas de la arquitectura anterior. Además aquí se configurará el enrutamiento entre las diferentes redes así como la publicación de servicios hacia el exterior.

➤ Ordenador DMZ:

Portátil Intel Core I5, 4GB RAM, 250GB HDD

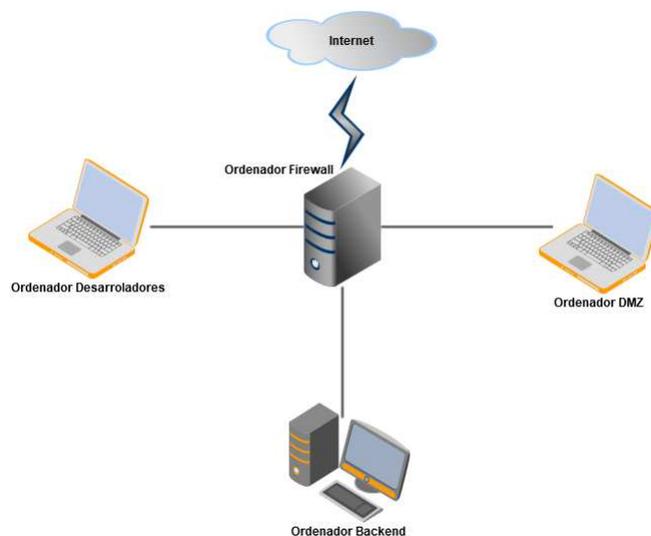
En este virtualizamos mediante PFSense todas las máquinas que irían incluidas en el DMZ de nuestra red, es decir aquellas que ofrecerán servicios al exterior y que verán sus puertos publicados hacia el exterior.

➤ Ordenador Desarrollo:

Portatil Intel Core I5, 4GB RAM, 250GB HDD

Este portátil contará con una instalación de Debian en su versión de Escritorio y es donde se realizará todo el despliegue del software del que harán uso final nuestros desarrolladores y demás empleados de la empresa de desarrollo, tales como IDEs de desarrollo, aplicaciones ofimáticas, etc.

El esquema organizativo de lo anteriormente descrito se presenta debajo mediante el siguiente gráfico:





Para simular la arquitectura Dual Homed lo que hemos hecho es añadir una tarjeta de red extra a nuestro Firewall, definiendo una de ellas como red DMZ con las consiguientes restricciones de acceso y de seguridad que esto representa que quedarán reflejadas en las tablas de NAT y FIREWALLING de nuestro PFSense.

En esta imagen se muestra el estado de nuestro laboratorio, incluyendo las máquinas arriba descritas con las interconexiones de red utilizadas.

2.2 Infraestructura software

La forma escogida para desplegar la infraestructura de software en los distintos ordenadores de los que disponemos es la siguiente.

En el Ordenador de DMZ desplegaremos máquinas con los siguientes servicios:

- Servidor WEB Apache
- DNS
- NTP
- Zimbra

En el ordenador correspondiente al Firewall simplemente dispondremos la distribución PFSense que cubrirá todas nuestras necesidades en esta zona.

En el ordenador correspondiente al Backend, realizaremos los siguientes despliegues:

- Proxmox VE
- Nagios
- OpenLDAP
- Openfire
- Mysql
- Tomcat
- Redmine
- Subversion

Y por último en el ordenador que se encarga de la parte de clientes, vamos a desplegar:

- Netbeans
- Eclipse
- Libreoffice

Como podemos observar la distribución elegida para desplegar el software va íntimamente ligada a la función que va a desarrollar el mismo dentro de la infraestructura de la empresa, se mantienen los servicios de publicación y comunicaciones con el exterior dentro de la zona desmilitarizada, mientras que se restringen los servicios que conforman la intranet de la empresa a la zona de backend o zona de seguridad, a la cual no se podrá acceder desde el exterior directamente si no es mediante una VPN, además todo el tráfico entre la zona desmilitarizada y la intranet estará controlado mediante nuestro Firewall que garantizará que solo se acceda a lo realmente necesario.

3. Soluciones en Software Libre adoptadas

En el siguiente apartado vamos a reflejar cuales han sido las soluciones en Software Libre que se han adoptado para la implementación de nuestra infraestructura. En muchos casos se han evaluado otras posibles opciones y seguramente habrá muchas soluciones que cumplan con los requerimientos, pero aquí se han escogido las soluciones que se muestran a continuación basándonos en criterios de versatilidad y seguridad sobre todo.

3.1 Sistemas operativos

Como sistemas operativos se planteaban muchísimas opciones, tantas como variedades de linux hay disponibles, sin embargo hemos optado por implementar las más estándar y conocidas, esto nos va a permitir gozar de multitud de paquetes, aplicaciones y tutoriales para las mismas.

Debian en concreto ha sido la distribución elegida, puesto que es una de las distribuciones más extendidas a nivel mundial en el ámbito empresarial, esto es debido probablemente a su gran estabilidad ya que para ellos es una de las distribuciones que más tarda en incluir los paquetes de software en sus repositorios dado los estrictos controles por los que pasan antes de ser incluidos, garantizando así la mínima posibilidad de fallo o vulnerabilidad en los mismos.

3.1.1 Debian 7



Debian GNU/Linux es la principal distribución Linux del proyecto Debian, que basa su principio y fin en el software libre, y fue creada por el proyecto Debian en el año 1993.

Los sistemas Debian actualmente usan el núcleo de Linux o de FreeBSD, sin embargo, se está trabajando para ofrecer Debian con otros núcleos, en especial con el Hurd, que es una colección de servidores que se ejecutan sobre un micronúcleo (como Mach) para implementar las distintas funcionalidades.

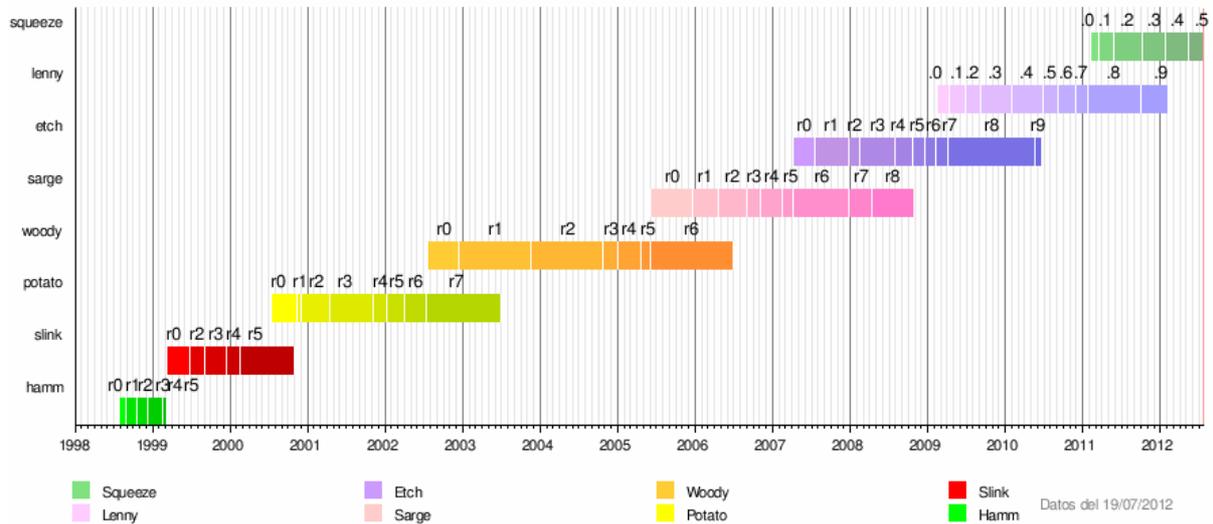
Nace como una apuesta por separar en sus versiones el software libre del software no libre. El modelo de desarrollo es independiente a empresas, creado por los propios usuarios, sin depender de ninguna manera de necesidades comerciales. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia. Muchas de las herramientas básicas, también libres, que lo completan vienen del proyecto GNU (GNU/Linux, GNU/kFreeBSD, GNU/Hurd).

Pese a que Debian cree en el software libre, hay casos en los que se quiere o necesita disponer de software que no sea libre en las máquinas, y siempre que sea posible, Debian respaldará ésto. Existe un número creciente de paquetes que tienen como única misión instalar software no libre en un sistema Debian.

Debian puede funcionar en casi todos los ordenadores personales, incluyendo muchos de los modelos más antiguos, pero existen algunas compañías que hacen difícil el soporte al no publicar las especificaciones de su hardware y que además proporcionan drivers que no son libres, y ésto puede ser un problema, ya que la compañía puede quebrar o dejar de dar soporte a ese hardware.

- En *Debian Unstable* es donde tiene lugar el desarrollo activo de Debian, es la rama que usan los desarrolladores del proyecto. La rama inestable de Debian siempre tiene como nombre en clave *Sid*.
- *Debian Testing* es la *versión en pruebas* de Debian. En ella se encuentran paquetes que han estado previamente en la *versión Inestable*, pero que contienen muchos menos fallos. La versión actual de Debian Testing es 'Jessie'.
- *Debian Stable* es la *versión normalizada*, cuenta con el apoyo del *Equipo de Seguridad de Debian* y es la recomendada para uso en producción, siendo **Wheezy** la versión estable actual. Hasta la fecha ha habido doce versiones estables.

Línea de tiempo de Debian GNU/Linux



En nuestro caso utilizamos Debian en las dos vertientes, tanto en su versión básica como sistema operativo para los servidores, como en su versión Desktop que es la que vamos a emplear para que los desarrolladores puedan realizar llevar a cabo su labor diaria.

Para la instalación de nuestra distribución utilizamos una imagen de Debian 7, concretamente esta imagen:

<http://cdimage.debian.org/debian-cd/7.3.0/amd64/iso-cd/debian-7.3.0-amd64-netinst.iso>

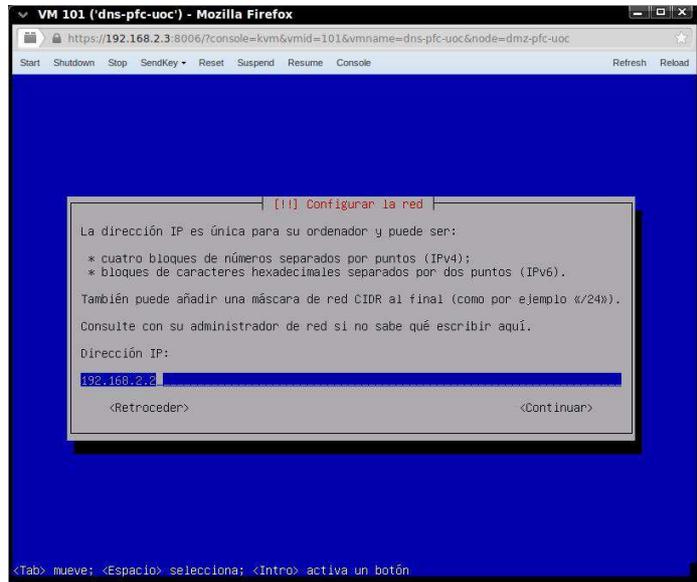


Y a partir de la misma podremos instalar tanto la versión para servidor como la versión para escritorio de nuestro sistema operativo.

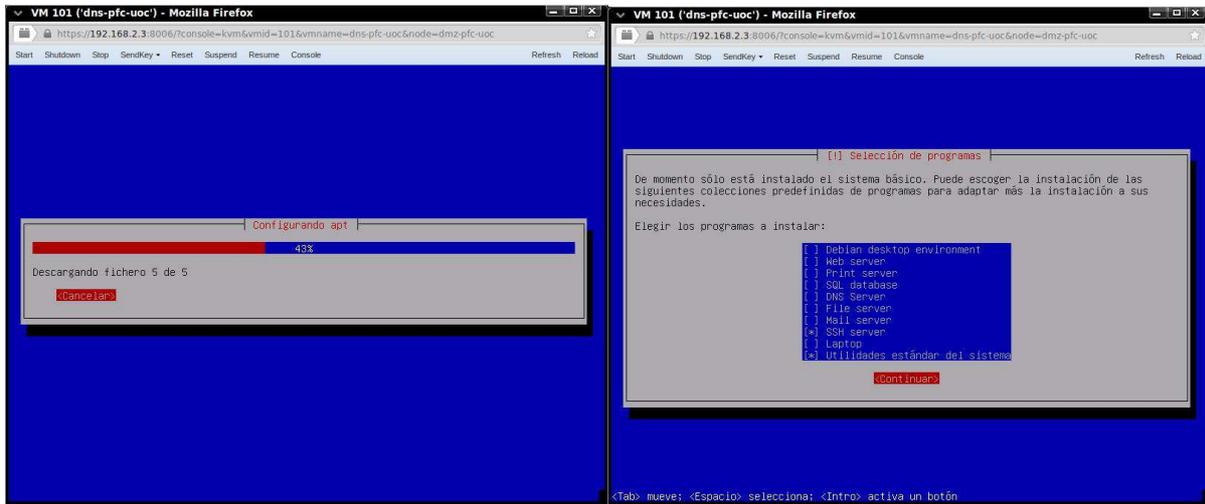
A continuación vamos a mostrar unos pantallazos del proceso de instalación: en una de nuestras máquinas virtuales:

Elegimos siempre la instalación manual sin entorno gráfico, lo que nos permitirá personalizar al máximo nuestra instalación.

En la pantalla de selección de DNS vamos a introducir la dirección de IP local del servidor de nombres de la red local de nuestra empresa, puesto que esto nos permitirá no solo acceder a internet, si no alcanzar las máquinas de nuestra red local de la empresa utilizando nombres en lugar de tener que recordar direcciones IP. Esto hará que los usuarios trabajen de una forma mucho más cómoda.



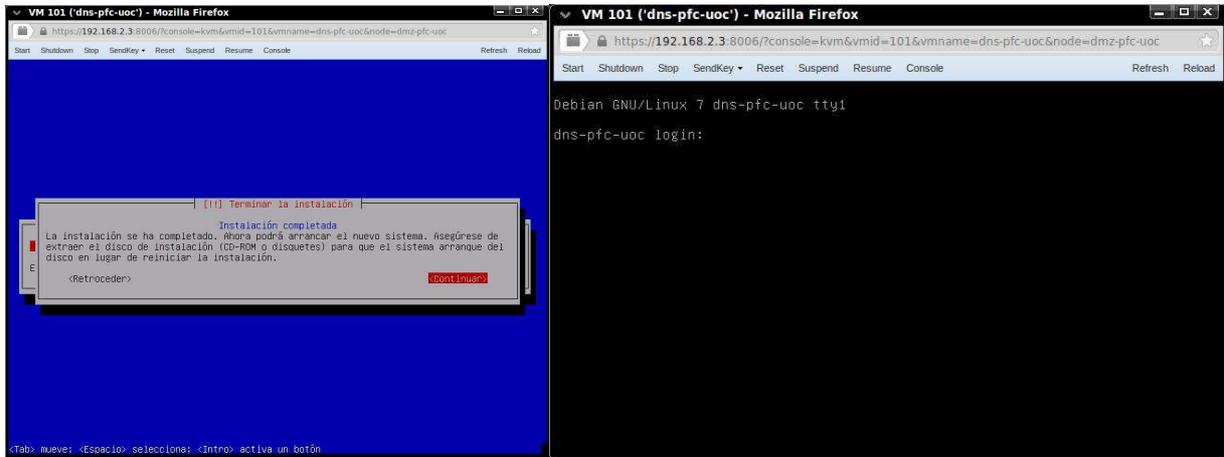
Tras las instalación del sistema base llegaremos a la parte en la que deberemos escoger la paquetería necesaria en función de las utilidad que queramos darle a la máquina concreta, en este caso, va a ser un servidor, que luego personalizaremos con alguno de los servicios por lo que lo único que instalaremos serán las utilidades básicas del sistema así como el servidor SSH que nos permitirá acceder y trabajar con el servidor de forma remota:



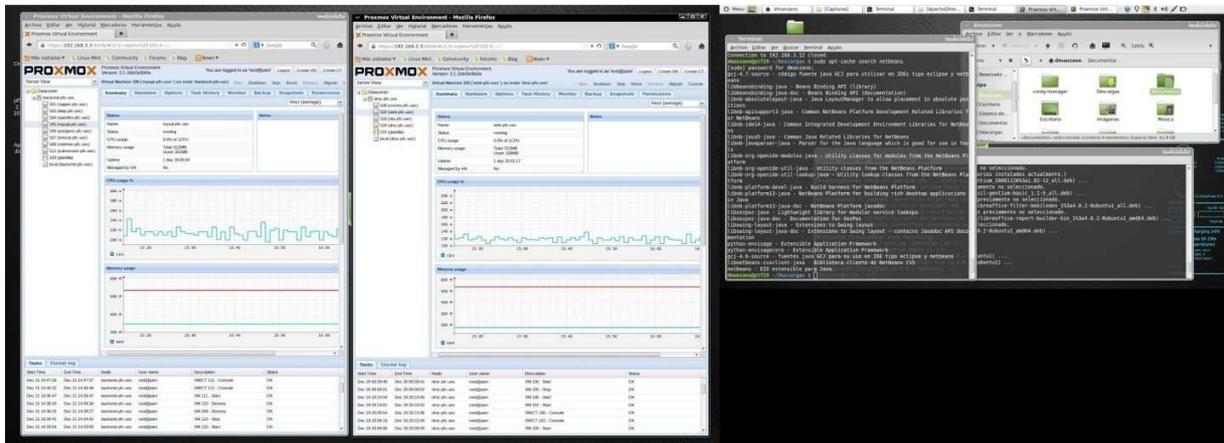
Y ya tenemos nuestra instalación completada, un sistema Debian 7.3 preparador para trabajar.

Implementación de GNU/Linux para una empresa de desarrollo de software

David Manzano Fiestas



Esta máquina en concreto será la que configuraremos como DNS por lo que ya le hemos asignado dicho nombre. Si en lugar de seleccionar la instalación Base escogemos instalar un entorno de escritorio, entonces acabaremos con un sistema que será el implementado en los equipos para todos los desarrolladores de la empresa. En la siguiente imagen podemos ver como luce el entorno Debian de escritorio que se utiliza para la simulación de cliente de la empresa y desde el cual se ha llevado a cabo toda la configuración, trabajo e implementación del resto de sistemas.



El sistema utiliza un sistema de escritorio extendido, por lo cual contamos dos pantallas para poder trabajar con mayor velocidad y comodidad. Gracias a la versatilidad de un sistema como es Debian, nuestro desarrolladores, podrán realizar con sus equipos cualquier tipo de tarea, tales como navegación, desarrollo, comunicación con el exterior.

Los repositorios del sistema nos van permitir instalar software de lo más variado, desde el potente editor Emacs, proveniente del corazón de GNU, hasta herramientas colaborativas y privativas, como Skype o Dropbox, pasando por las que se comentarán más adelante que serán las herramientas de desarrollo que serán de las que harán uso nuestros desarrolladores para su labor diaria.

3.1.2 FreeBSD:



Está basado en las CPU de arquitectura Intel, incluyendo procesadores 386, 486 (versiones SX y DX), y Pentium. También funciona en procesadores compatibles con Intel como AMD y Cyrix. Actualmente también es posible utilizarlo hasta en once arquitecturas distintas como Alpha, AMD64, IA-64, MIPS, PowerPC y UltraSPARC.

FreeBSD está basado en la versión 4.4 BSD-Lite del Computer Systems Research Group (CSRG) de la University of California, Berkeley siguiendo la tradición que ha distinguido el desarrollo de los sistemas BSD. Además del trabajo realizado por el CSRG, el proyecto FreeBSD ha invertido miles de horas en ajustar el sistema para ofrecer las máximas prestaciones en situaciones de carga real.

En nuestro caso particular, hemos utilizado esta versión para implementar el Firewall de nuestra empresa, utilizando la distribución PfSense que se especializa en Firewall. Aunque FreeBSD no es puede denominar como GNU/Linux en sí puesto que no contiene el kernel Linux ni las herramientas de GNU propiamente, si se basa en la misma filosofía de sistema operativo libre, pero con el añadido de que la funcionalidad de PFSense está especializada en software de Firewall y control, cosa que no habría pasado si utilizásemos cualquier otro software.

Pfsense es una distribución de código abierto basada en FreeBSD que se utiliza como firewall y router. Puede ser configurada y actualizada a través de una interfaz web y no requiere conocimiento del sistema FreeBSD para manejarlo, además está dotado de un gestor de paquetes para poder ampliar sus funcionalidades, y existen unos setenta módulos disponibles.

Gracias a PfSense seremos capaces de gestionar la red completa de nuestra organización, este punto nos va a permitir establecer las diferentes redes, realizar el enrutamiento entre ellas, controlar el tráfico, mapear puertos al exterior, etc....

Aunque en un principio se contempló la idea de virtualizar este aplicativo, finalmente y por razones de eficiencia hemos terminado implementado dicho sistema en una máquina física. Esto nos va a permitir una configuración más rápida del entorno, además de una simulación más real del mismo, puesto que lo normal en cualquier

empresa es que el firewall esté contemplado en una máquina física la cual, con sus diferentes interfaces de red, es capaz de segmentar la red de la empresa y configurar muchos de los servicios críticos de la misma. Podríamos decir que este firewall se configuraría como uno de los pilares básicos de funcionamiento de toda la red de la empresa, sin el cual sería imposible conformar la infraestructura.

Para ello hemos utilizado un ordenador Athlon AMD64 con 2Gb de ram y 70Gb de disco duro. En nuestro caso es más que suficiente para la red que estamos simulando , y con la carga a la que lo hemos sometido apenas hemos conseguido que dicha máquina sufra o mantenga algún tipo de carga elevada de trabajo.

Sin embargo y en función de las pruebas realizadas, hemos podido constatar como el número de conexiones abiertas influye directamente sobre la carga del procesador, y en el caso de que el número de desarrolladores o de servicios mantenidos por dicha máquina fuese creciendo con el tiempo es posible que para evitar cuellos de botella se tuviese que sustituir dicho ordenador por alguno con capacidades superiores de memoria y procesador.

A continuación mostramos el proceso de instalación en en nuestro laboratorio. Las imágenes están tomadas con cámara externa de fotos puesto que el sistema no está virtualizado y nos era imposible realizar capturas sobre la pantalla directamente hasta el momento de conectarnos al mismo mediante interfaz web.



A continuación podemos ver una imagen de la máquina que actuará en nuestro sistema simulando el firewall de nuestra empresa:

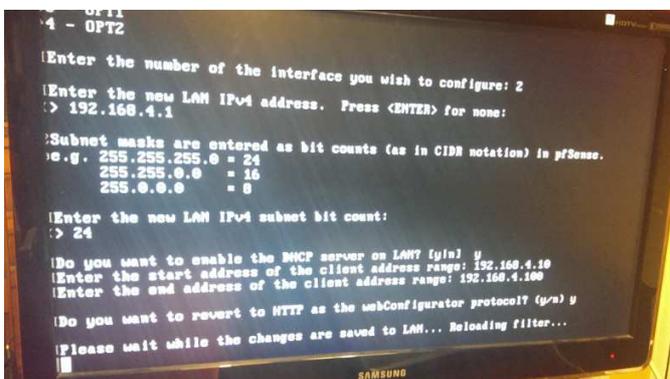
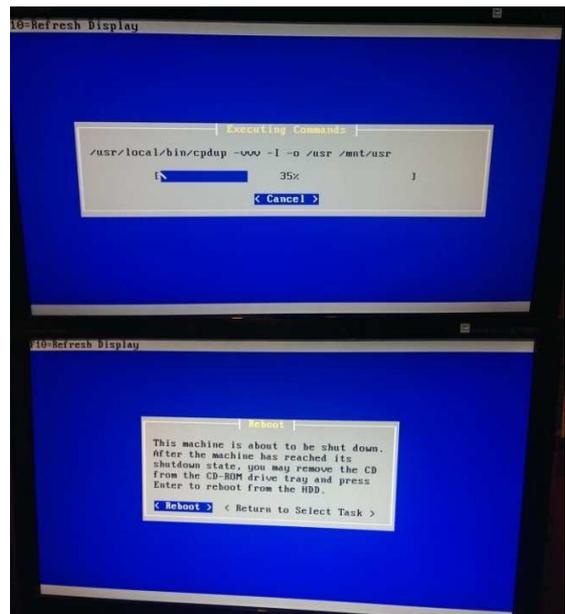
En la imagen podemos ver como se han agregado dos tarjetas de red PCI, son estas tarjetas las que van a realizar las funciones de diferentes puntos de entrada de las redes de nuestra empresa virtual.

En esta imagen podemos ver un detalle de dichas tomas de red:



La toma superior proviene directamente del router de Telefónica, es decir, ese actuará como la WAN de entrada a nuestro sistema, y las tomas inferiores serán las 3 subredes distintas, una para el DMZ, otra para el BACKEND y la otra en la que se ubicarán nuestros desarrolladores y donde se encuentra ubicado el ordenador que se utiliza para configurar todo el sistema.

Con esta infraestructura comenzamos la instalación o configuración básica del sistema PfSense:



Tras esta instalación y el primer arranque es el momento de configurar nuestras diferentes interfaces de red.

Hasta dejarlas como se muestra a continuación:

Ahí ya podemos ver como cada interfaz tiene su IP y la hemos identificado según la función que va a realizar. Ahora es el momento de conectar nuestro portatill de desarrollo a la toma lan (192.168.4.1) y dejar que nos conceda una dirección mediante DHCP para que podamos acceder a través de la IP antes mencionada a la interfaz web.

```
Starting package darkstat...done.
Starting package Lightsquid...done.
Starting package bandwidthd...done.
Starting /usr/local/etc/rc.d/proxy_monitor.sh...done.
Bootup complete

FreeBSD/amd64 (firewall-tfc-uoc.dmanzano.org) (ttyv0)

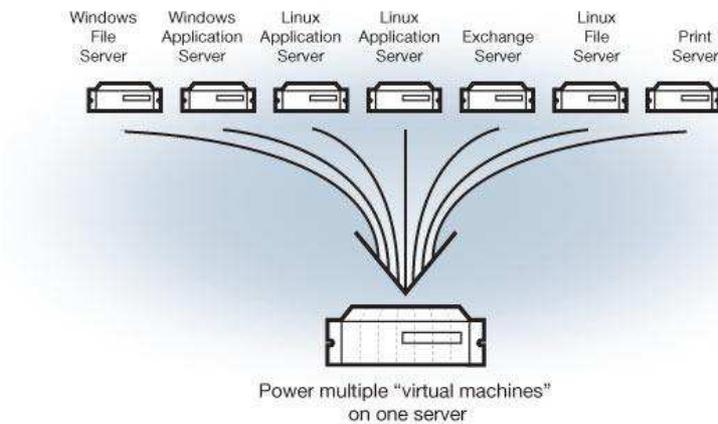
*** Welcome to pfSense 2.0.1-RELEASE-pfSense (amd64) on firewall-tfc-uoc ***

WAN_INTERNET (wan)      -> r10      -> 192.168.1.10
USUARIOS (lan)         -> hme0     -> 192.168.4.1
DMZ (opt1)             -> hme1     -> 192.168.2.1
BACKEND (opt2)        -> hme2     -> 192.168.3.1

0) Logout (SSH only)          8) Shell
1) Assign Interfaces          9) pfTop
2) Set interface(s) IP address 10) Filter Logs
3) Reset webConfigurator password 11) Restart webConfigurator
4) Reset to factory defaults 12) pfSense Developer Shell
5) Reboot system             13) Upgrade from console
6) Halt system               14) Disable Secure Shell (sshd)
7) Ping host

Enter an option: █
```

3.2 Virtualización



Mediante software la **virtualización** creamos una versión virtual de algún recurso tecnológico (un sistema operativo, un dispositivo de almacenamiento...). El software ejecutado en estas máquinas virtuales está separado de los recursos de hardware subyacentes.

Por ejemplo, un ordenador que está ejecutando Microsoft Windows puede alojar una máquina virtual que

parezca un ordenador con el sistema operativo Ubuntu Linux.

La máquina física se conoce como "host" y el sistema operativo de la máquina virtual se conoce como "virtual machine" o "guest".

Los diferentes tipos de virtualización de hardware incluyen:

- Virtualización completa: simulación casi completa del hardware actual que permita al software, que normalmente consiste en un sistema operativo invitado, ejecutarse sin modificar.

- Virtualización parcial: No todo el entorno de destino se simula. Algunos programas invitado pueden necesitar modificaciones para funcionar en este entorno virtual.
- Paravirtualización: No se simula un entorno de hardware, sin embargo, los programas invitado se ejecutan en sus propios dominios aislados, como si se estuvieran ejecutando en un sistema separado. Los programas invitado deben ser modificados específicamente para funcionar en este entorno.

Con la virtualización, varios sistemas operativos se pueden ejecutar en paralelo en una unidad de procesamiento central única (CPU).

Usando la virtualización, una empresa puede administrar mejor las actualizaciones y los cambios rápidos en el sistema operativo y las aplicaciones sin interrumpir al usuario.

En nuestro caso la virtualización es clave para todo el funcionamiento de nuestro entorno de laboratorio, puesto que la mayor parte de él está simulado bajo virtualización, sin ésta habría sido imposible la realización de este proyecto puesto que deberíamos haber contado con muchos más recursos hardware de los realmente utilizados.

3.2.1 Contenedores con OpenVZ



OpenVZ

OpenVZ es una tecnología de virtualización a nivel de sistema operativo basado en el kernel y el sistema operativo de Linux, es una de las mejores soluciones para virtualizar en Linux. Al compartir el contenedor el mismo kernel del sistema operativo anfitrión, se pueden

ahorrar muchos recursos, ya que en virtualización solo se usa el 1% o 2% de los recursos de la CPU. También por este motivo, cada contenedor, que mientras sea Linux puede ser cualquier cosa (Ubuntu, debian...), se va a ver y comportar como un sistema Linux normal y sin hacer modificaciones o ajustes para correr sobre OpenVZ.

OpenVZ tiene una administración de recursos excelente, permitiendo que se compartan todos los recursos disponibles del sistema anfitrión entre todos los contenedores, también garantiza la Calidad de Servicio, provee un aislamiento de recursos y recolecta información de uso para monitorear el estado del sistema.

Cuatro componentes principales de esta administración de recursos:

- Dos niveles de cuota de disco
- Planificador de Procesador justo
- Planificador de Entrada/Salida
- Beancounters de usuario

Unas de las características más importantes de OpenVZ son el checkpointing y la migración en vivo, que hacen posible poder migrar los contenedores de un servidor físico a otro sin tener que apagar el contenedor y reiniciarlo luego.

3.2.2 Virtualización completa con KVM



Es una solución para Linux en hardware x86 que contiene extensiones de virtualización, es un hypervisor único.

Usando KVM, se pueden ejecutar múltiples máquinas virtuales corriendo imágenes sin modificar de Linux o Windows. Cada máquina virtual tiene un hardware virtualizado privado (tarjeta de red, disco, tarjeta gráfica, etc). Una de las ventajas es que el entorno virtualizado se puede beneficiar de todo el trabajo realizado en el mismo kernel de Linux.

Un proceso normal de Linux dispone de dos modos de ejecución: de Kernel y de usuario. KVM agrega un tercer modo, el modo de invitado.

3.2.3 ProxmoxVE



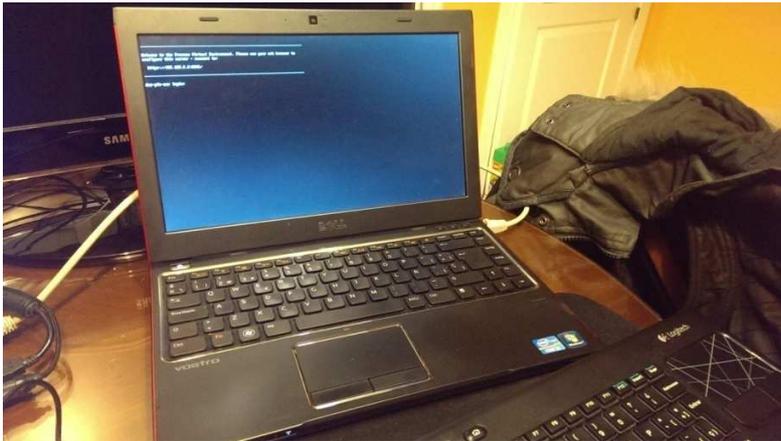
ProxmoxVE es una completa **plataforma de virtualización** basada en sistemas de código abierto que permite la virtualización tanto sobre **OpenVZ** como **KVM** y gestiona máquinas virtuales, almacenamiento, redes virtuales y clusters de alta disponibilidad.

Permite la migración en vivo de máquinas virtuales, clustering de servidores, backups automáticos y conexión a un NAS/SAN con NFS, iSCSI, etc...

Como utiliza OpenVZ se puede cambiar tanto la memoria RAM como el espacio en disco asignados en tiempo real sin necesidad de reiniciar el sistema.

Este sistema es posiblemente el corazón de nuestra empresa de desarrollo, la potencia de la que dota el sistema a nuestro entorno es total, ya que este se encarga de virtualizar y gestionar todas las copias de seguridad de los entornos de la empresa. La decisión sobre qué sistemas o no se deben virtualizar se deberán tomar en función de la criticidad y carga de los mismos. En nuestro caso vamos a virtualizar todos los sistemas, esto nos va a garantizar una capacidad de respuesta segura ante problemas en algún servidor, y un mantenimiento sencillo de las copias de seguridad.

Aunque en un principio se consideró la posibilidad de virtualizar estas máquinas para nuestra simulación finalmente se ha terminado trabajando nativamente con el sistema Proxmox sobre las máquinas, esto nos ha garantizado velocidad y facilidad en la configuración sobre todo a nivel de red, puesto que los enlaces serán físicos y no virtuales y estarán conectados directamente a nuestro firewall.



Las dos máquinas de virtualización que utilizaremos son las siguientes:

La primera un ordenador portátil Dell con un procesador i5 y 4GB de ram.

Este lo podemos ver ya con la distribución proxmox desplegada. Debido a su menor potencia, este lo utilizaremos para virtualizar las máquinas que situaremos en nuestro DMZ, que serán menos que las que habrá en nuestro Backend.

El otro ordenador que utilizamos para virtualización será un i7 2600K con 8GB de RAM al cual se le ha sustituido el disco duro por uno de 160Gb que emplearemos específicamente para esta labor.

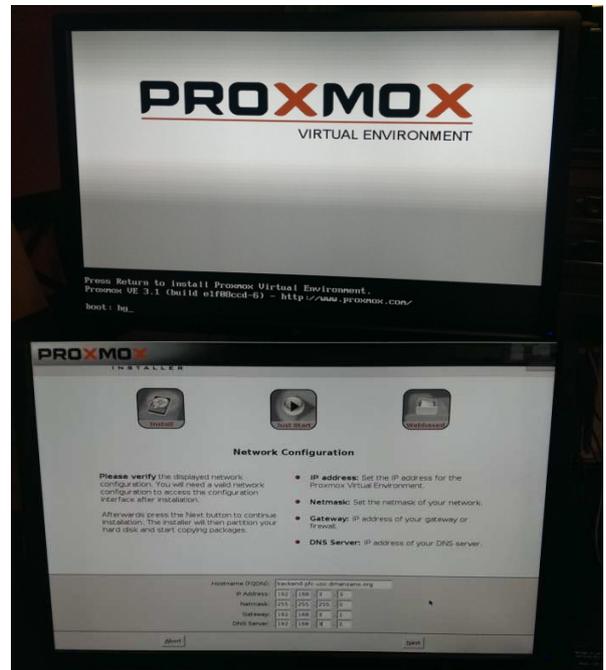
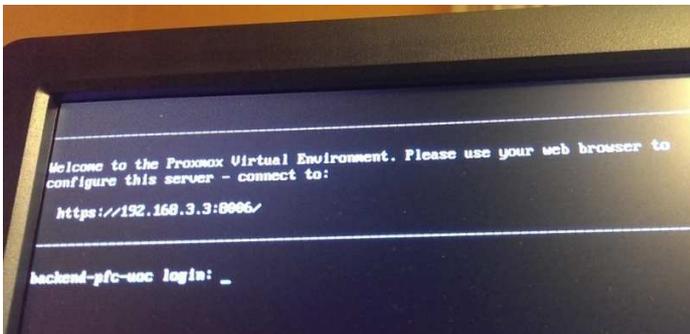


Es este vamos a desplegar toda la parte de backend de nuestro sistema. Gracias a la velocidad del procesador incluido el cual se encuentra overclockeado hasta los 5Ghz, disponemos de potencia suficiente como para albergar todas las máquinas virtuales que vamos a crear sin que notemos un decremento del rendimiento del mismo.

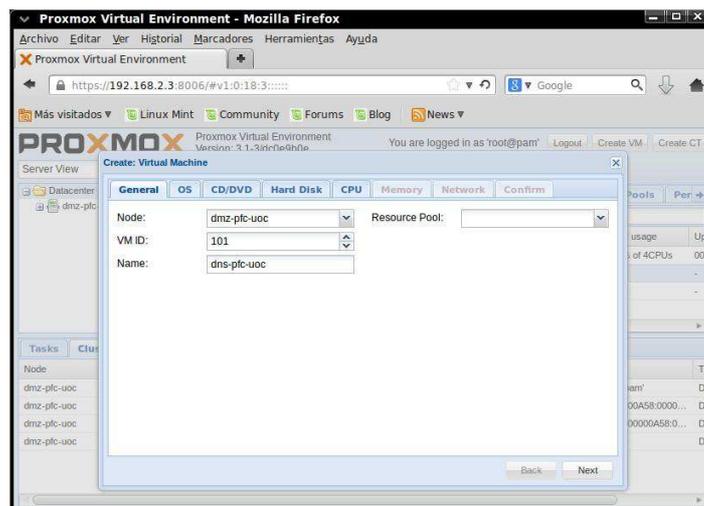
A continuación vamos a mostrar un proceso de instalación del entorno proxmox en uno de los servidores:

Implementación de GNU/Linux para una empresa de desarrollo de software

David Manzano Fiestas



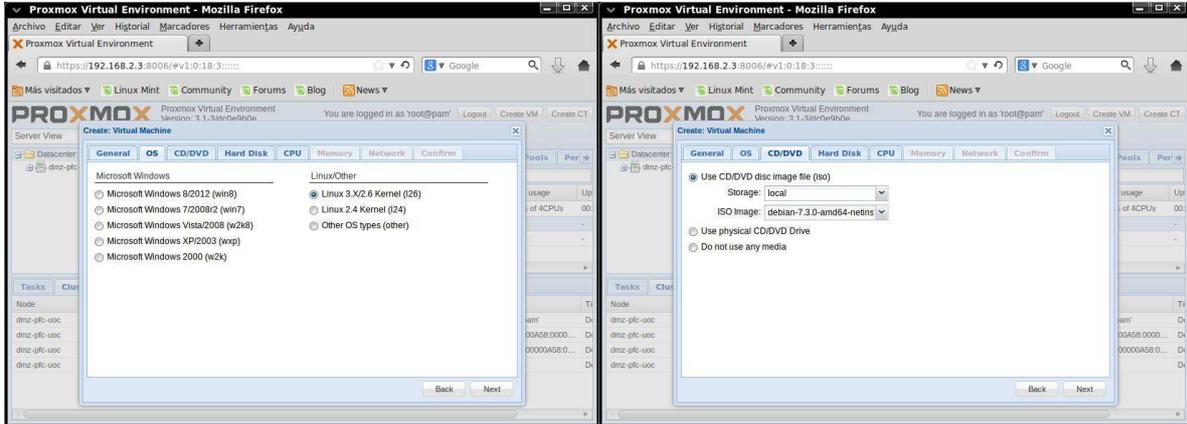
En concreto es la instalación en nuestro servidor de Backend. Una vez completado, podemos ver como se nos ofrece conectar desde una interfaz web que será la que utilizaremos para administrar el mismo, y si conectamos a través de la interfaz web ya tenemos nuestro sistema esperando para trabajar con él. Una vez tenemos el sistema desplegado en ambos ordenadores anfitriones es el momento de empezar a configurar las diferentes máquinas virtuales. Para ello y usando la interfaz web podemos mostrar el proceso de generación de una máquina virtual: Primero deberemos rellenar los datos generales de la máquina que estamos creando:



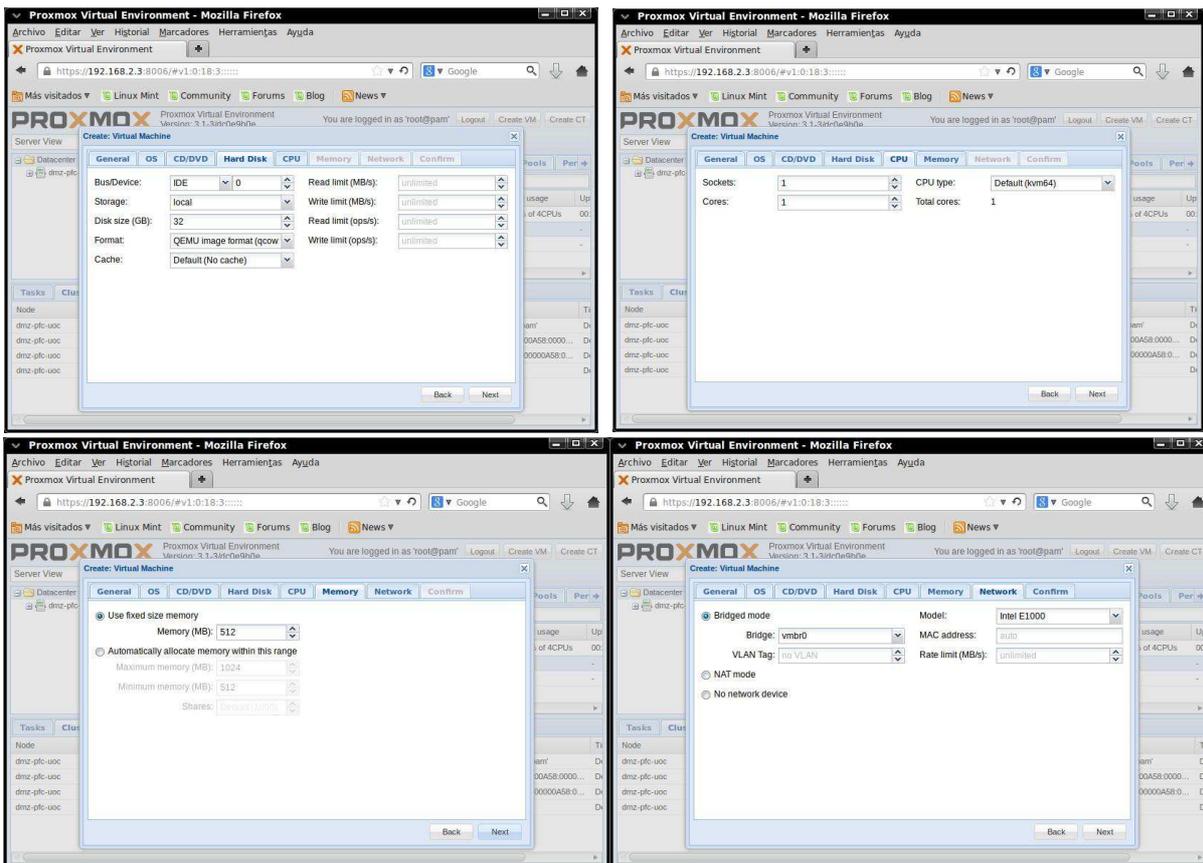
Implementación de GNU/Linux para una empresa de desarrollo de software

David Manzano Fiestas

Luego seleccionaremos el tipo de sistema operativo que vamos a albergar en la misma y definiremos el origen de instalación que utilizaremos:



El disco y CPU que queremos asignar a dicha máquina:



Una vez terminado podremos arrancar nuestra máquina virtual y realizar la instalación de nuestro sistema Debian como si de una máquina normal se tratase utilizando una consola de VNC que nos ofrece el mismo proxmox y que nos permite conectarnos a las máquinas y disponer de una pantalla virtual de la misma así como usar nuestro teclado y ratón sobre la máquina. Y todo esto a través de la interfaz web.

Proxmox ofrece además muchas otras posibilidades como migraciones vivo, cluster, alta disponibilidad y copias de seguridad, de las que hablaremos en otro de los apartados de este trabajo.

3.3 Seguridad

3.3.1 Pfsense



Además de ser un potente y flexible cortafuegos, y una plataforma de enrutamiento, incluye una larga lista de características relacionadas y un sistema de paquetes que permite mayor capacidad de expansión sin necesidad de añadir potenciales vulnerabilidades de seguridad a la distribución base.

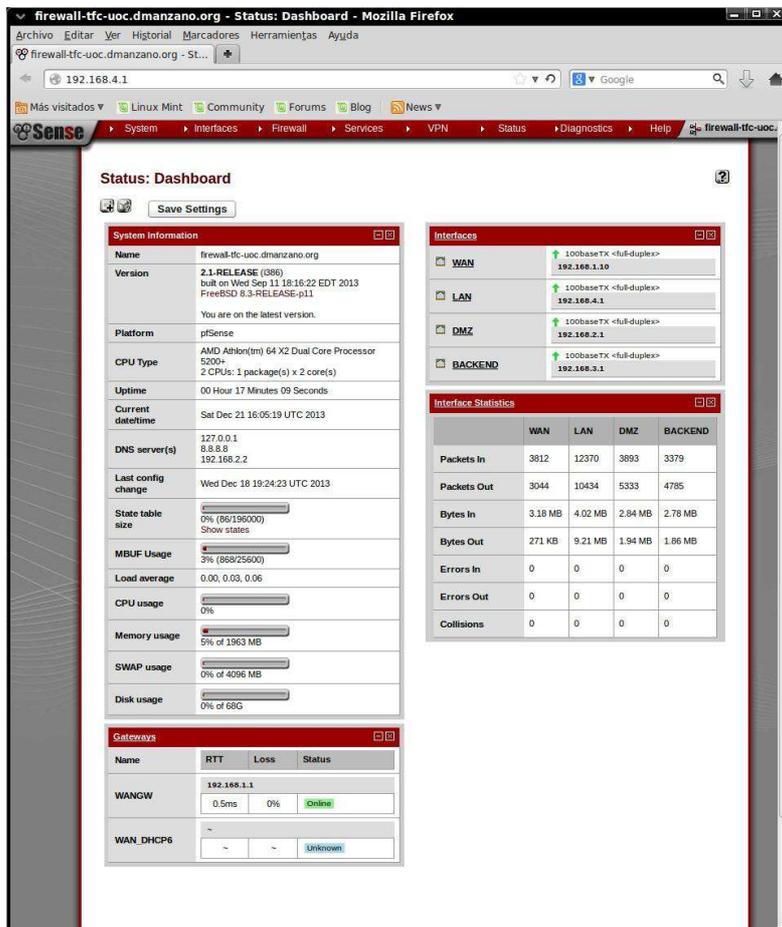
Este proyecto se inició en 2004, enfocado hacia instalaciones completas de PC en lugar del enfoque hardware embebido de mOnOwall. Ahora también ofrece una imagen integrada para las instalaciones basadas en Compact Flash. Desde entonces, los dos proyectos han divergido significativamente.

Características:

- Firewall
- Tabla de estados
- Network Address Translation (NAT)
- Balanceo de carga
- Portal Cautivo
- uPnP
- VPN que puede ser implementada en IPsec, OpenVPN y en PPTP
- Servidor DHCP
- Servidor PPPoE
- Servidor DNS
- Paquete de soporte “push-button install” para extensiones

Pfsense se convertirá en una de las piezas clave de nuestra configuración. Este se va a encargar de gestionar muchos de los servicios que estamos implementando, así como en él estableceremos los mecanismos de seguridad que evitarán que haya accesos no autorizados a los servicios desplegados en los diferentes segmentos de red.

En la parte de Pfsense explicamos cómo establecemos los diferentes sistemas de seguridad en el mismo. Y para ello accederemos a través de una interfaz web que el sistema nos suministra y que nos permitirá la configuración del mismo de una forma bastante intuitiva y cómoda.



Una vez que accedemos, procederemos a configurar las diferentes redes con sus reglas que nos permitirán todo el funcionamiento del sistema, ahora adjuntamos una serie de pantallazos de cuales son las reglas que se han incrustado en el firewall.

La primera son las reglas en la WAN que nos permitirán navegar por internet así como dejar paso de todos los servicios que publicamos desde nuestro DMZ al exterior:

Firewall: Rules

Tab: Floating | WAN | LAN | DMZ | BACKEND | PPTP VPN

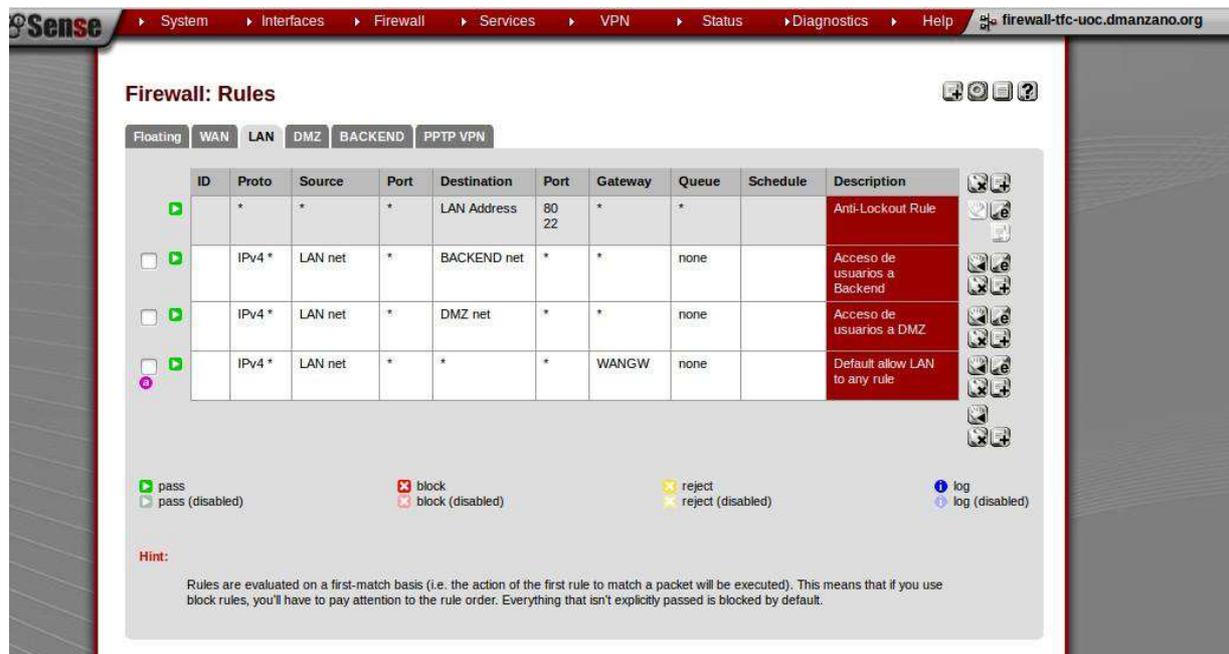
ID	Proto	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
<input checked="" type="checkbox"/>	*	RFC 1918 networks	*	*	*	*	*	*	Block private networks
<input checked="" type="checkbox"/>	*	Reserved/not assigned by IANA	*	*	*	*	*	*	Block bogon networks
<input type="checkbox"/>	IPv4 TCP	*	*	WAN address	1723 (PPTP)	*	none		Permitido acceso VPN
<input type="checkbox"/>	IPv4 TCP	*	*	192.168.2.4	123 (NTP)	*	none		Publicación del servidor horario a internet
<input type="checkbox"/>	IPv4 TCP	*	*	19.168.2.6	25 (SMTP)	*	none		Publicación del SMTP al exterior
<input type="checkbox"/>	IPv4 TCP	*	*	192.168.2.2	53 (DNS)	*	none		Publicación DNS al exterior
<input type="checkbox"/>	IPv4 TCP	*	*	192.168.2.6	143 (IMAP)	*	none		IMAP de Zimbra Publicado
<input type="checkbox"/>	IPv4 TCP	*	*	192.168.2.5	80 (HTTP)	*	none		Publicación HTTP al exterior desde la DMZ
<input type="checkbox"/>	IPv4 TCP	*	*	*	*	WANGW	none		

pass
 pass (disabled)
 block
 block (disabled)
 reject
 reject (disabled)
 log
 log (disabled)

Hint:
 Rules are evaluated on a first-match basis (i.e. the action of the first rule to match a packet will be executed). This means that if you use block rules, you'll have to pay attention to the rule order. Everything that isn't explicitly passed is blocked by default.

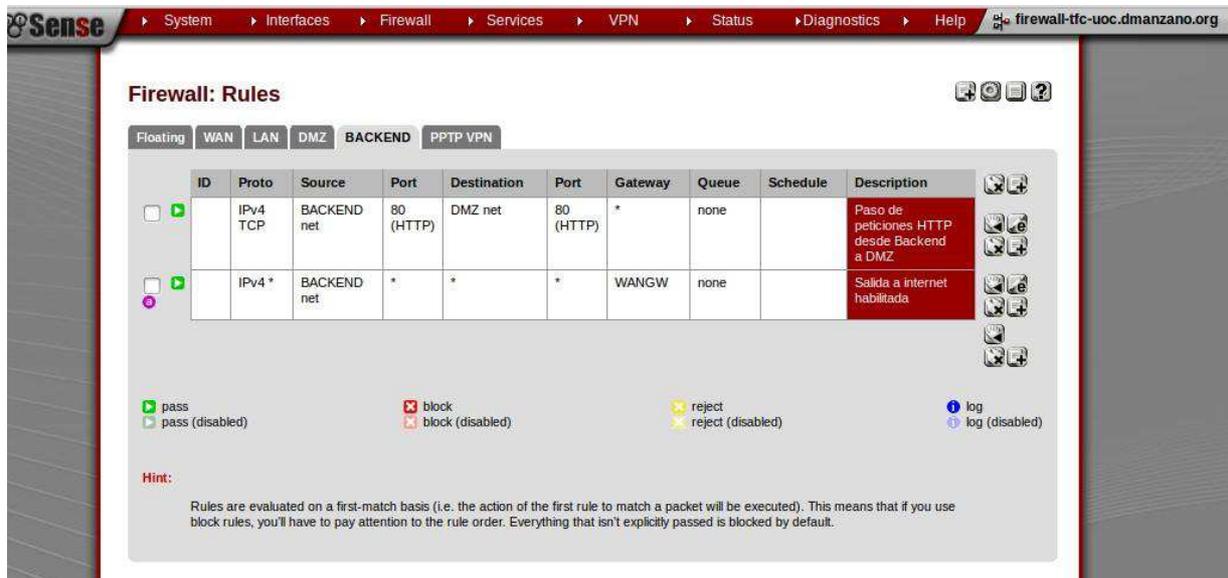
Además aquí añadimos el gateway por defecto de nuestra red. Por muchas de esas reglas se han ido añadiendo a posteriori cuando se definieron las ips de los servicios que se encontraban desplegados en cada sistema.

La red de usuarios es probablemente la más sencilla de configurar, puesto que estos van a tener acceso a todas las partes de nuestro sistema, por supuesto si se quisiesen aumentar los niveles de seguridad, se podrían haber definido distintas Vlans, y cada una de ellas habría tenido un acceso limitado a según que partes de la infraestructura, pero por simplificación todos nuestros usuarios de la red local tienen acceso completo a todo el sistema, así las reglas que definen su capacidad de comunicación quedan como sigue:

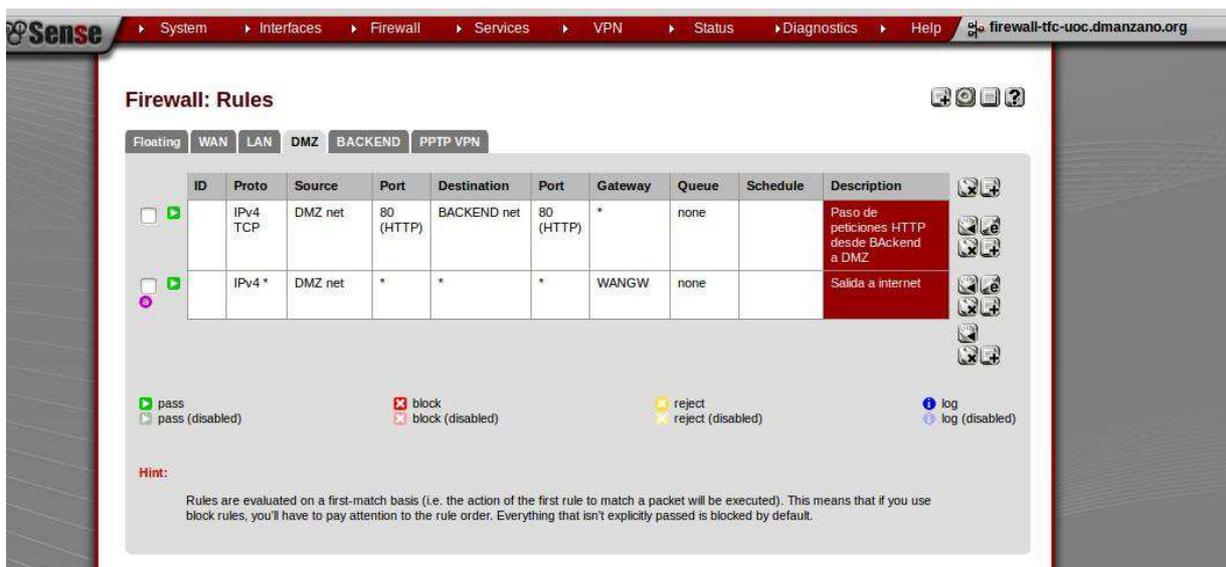


Como se puede observar, los usuarios tienen salida a internet y acceso a nuestras dos subredes. Así vamos a poder trabajar con ambas sin problemas desde nuestra estación de desarrollo.

Para las máquinas situadas en el Backend, veremos como ya tienen una conexión mucho más limitada. De hecho, estas podrán conectar con internet, y a través del puerto 80 en protocolo TCP con nuestro DMZ, esto nos servirá para publicar servicios web ubicados en nuestro backend al exterior, utilizando un proxy del DMZ, de este modo mantenemos seguro nuestro backend aislándolo del exterior. Por lo demás estas máquinas no podrán acceder a la red de usuarios ni al resto de la red DMZ. Un modelo de seguridad ampliada sería el de imitar la conectividad de las mismas a los repositorios de paquetería de la distribución Debian, de modo que estas no tuviesen ninguna capacidad de conectividad más allá, pero por simplificación se vuelve a dejar conectividad desde las máquinas hacia el exterior, pero nunca en el sentido inverso.



Y por último el DMZ, desde el DMZ podemos realizar conexiones con el exterior, y hacia el puerto 80 de nuestro BACKEND, y poco más. Esto nos garantiza que, puesto que estas máquinas son las que se encuentran publicadas al exterior y son más susceptibles de ataque, si algún intruso consiguiese hacerse con el control malintencionado de una de ellas, nunca pudiese acceder al resto de nuestra red corporativa, donde se encuentran nuestros servicios de backed que posiblemente contengan información sensible.



Este proceso resume la configuración básica que hemos llevado a cabo sobre nuestro firewall de empresa.

La potencia que nos ofrece esta distribución nos va a permitir realizar muchas más funcionalidad y ofrecer muchos otros servicios, como por ejemplo la VPN que veremos en un apartado posterior. Pero gracias a esta configuración básica ya tenemos definidos los mecanismos de control y seguridad básicos para el correcto funcionamiento operativo de nuestra empresa.

3.3.2 Copias de seguridad

Para la gestión de las copias de seguridad no se va a implementar ningún software específico, puesto que mediante la virtualización vamos a garantizar que se producen copias completas de las máquinas que soportan nuestra infraestructura. El seguir este esquema presenta una ventaja fundamental y es que en caso de desastre, la recuperación del servicio es mucho más rápida que se se tratase de una copia de seguridad convencional.

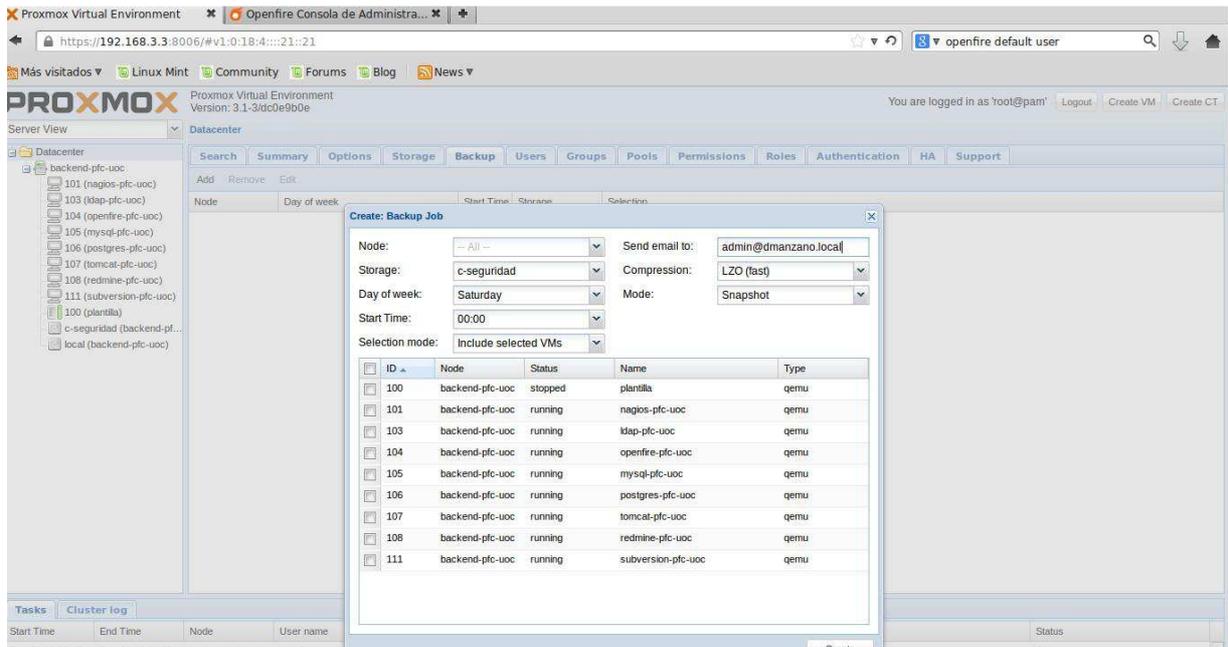
Pongámonos en el caso de que se produce un fallo de disco en nuestro servidor de correo, si el servidor de correo es físico, para poder realizar la recuperación de dicho servidor tendríamos que montar una nueva máquina desde partiendo desde cero, en la cual instalásemos un nuevo servidor de correo y a la cual una vez terminado el despliegue tendríamos que incorporar los datos de nuestro servidor de correo antiguo. Como podemos suponer, este proceso es bastante tedioso y si no se tiene bien documentado y preparado puede provocarnos muchos problemas y quebraderos de cabeza, como podría ser la versión de dicho servidor que puede no coincidir con la versión de la que nosotros disponemos e incluso que los datos no fuesen compatibles, o librerías del sistema del servidor antiguo que en el nuevo no se han instalado y provocan algún tipo de fallo.

En nuestro caso al virtualizar el servidor de correo, si se produce un fallo en el disco del servidor, bastará con coger una copia de nuestro almacén de copias de seguridad y restaurar esa copia en un servidor de virtualización cuyos discos no presenten esa problemática, y si el fallo se produce en el disco virtual de nuestro servidor de correo, también podremos recuperar esa imagen anterior y copiar el contenido del disco a un nuevo disco virtual exento de dicho fallo.

Esto no quiere decir que la virtualización sea siempre la solución a seguir, puesto que es probable que en sistemas con una alta carga en determinados servicios, la virtualización puede actuar como cuello de botella para los servicios que se encuentran alojados en la misma, pero en nuestro caso de una pequeña empresa de desarrollo no nos vamos a encontrar con dicho problema por lo que las copias de seguridad quedarán completamente gestionadas mediante el servidor de virtualización.

La gestión de las copias de seguridad se lleva a cabo mediante la interfaz de Proxmox que se muestra de la siguiente manera:

Si accedemos a través de la web a la pantalla de definición de máquinas virtuales veremos como nos aparece la opción de realizar copias de seguridad y nos da a escoger sobre que máquinas queremos realizar dichas copias, el horario y demás opciones:



Y aquí ya podemos ver el sistema definido:



Como ejemplo nosotros hemos configurado una copia semanal para los sábados a las 00:00, aunque en la realidad esto no sería viable, estas copias deberían llevarse a cabo en función de la criticidad de cada máquina siguiendo su propio patrón temporal. Así nos encontraríamos con máquinas como la del correo electrónico que posiblemente

necesitase de una copia diaria, o máquinas estáticas como la que mantiene la web de la empresa, que si no sufre modificaciones si podríamos realizar copias semanales de la misma.

Gracias a este sencillo mecanismo de virtualización podríamos tener cubierta la necesidad de mantener copias de seguridad de todos nuestros servicios para que en caso de desastre pudiésemos recuperarlos lo antes posible minimizando la posible pérdida de información.

3.3.3 Monitorización con Nagios



Cada día más las empresas tienen presente que para que sus redes y sus sistemas funcionen de forma correcta es necesario llevar a cabo una monitorización detallada de todo para poder detectar fallos lo antes posible, para prevenir en algunos casos

que éstos afecten a la infraestructura de la compañía. En nuestro caso la monitorización es crítica, puesto que nos encontramos ante sistemas de los cuales depende el funcionamiento íntegro de la empresa. Nagios es uno de los sistemas más extendidos dentro del mundo de la monitorización, fundamentalmente por la versatilidad y la capacidad de adaptación a cualquier entorno que posee.

Algunas de las características principales de Nagios son:

- Monitorización de servicios de red (SMTP, POP3, HTTP, SNMP...)
- Monitorización de los recursos de sistemas hardware (carga del procesador, uso de discos, estado de puertos...)
- Independencia de sistemas operativos
- Interfaz web integrada
- Posibilidad de monitorización remota mediante túneles SSL cifrados o SSH
- Posibilidad de programar plugins específicos para nuevos sistemas
- Ampliación de funciones mediante el uso de complementos
- Envío de notificaciones por correo electrónico o sms
- Asignación de roles para los usuarios
- Grabación y rotación automática de los estados de los elementos monitorizados en archivos logs
- Creación de eventos

Los objetos monitorizados por Nagios se dividen en dos categorías:

- Hosts
- Servicios

Cada servicio está asociado al Host en que se ejecuta, y ambos pueden agruparse.

Las ventajas de Nagios son:

- Flexibilidad por un sistema de definición de objetos y a unos pocos tipos de objetos (Comandos, Periodos de Tiempo, Contactos y Grupos de Contactos, Host, Servicios, Escalado de hosts y servicios)
- Sistema de dependencias
- Posibilita planificar interrupciones en servicios

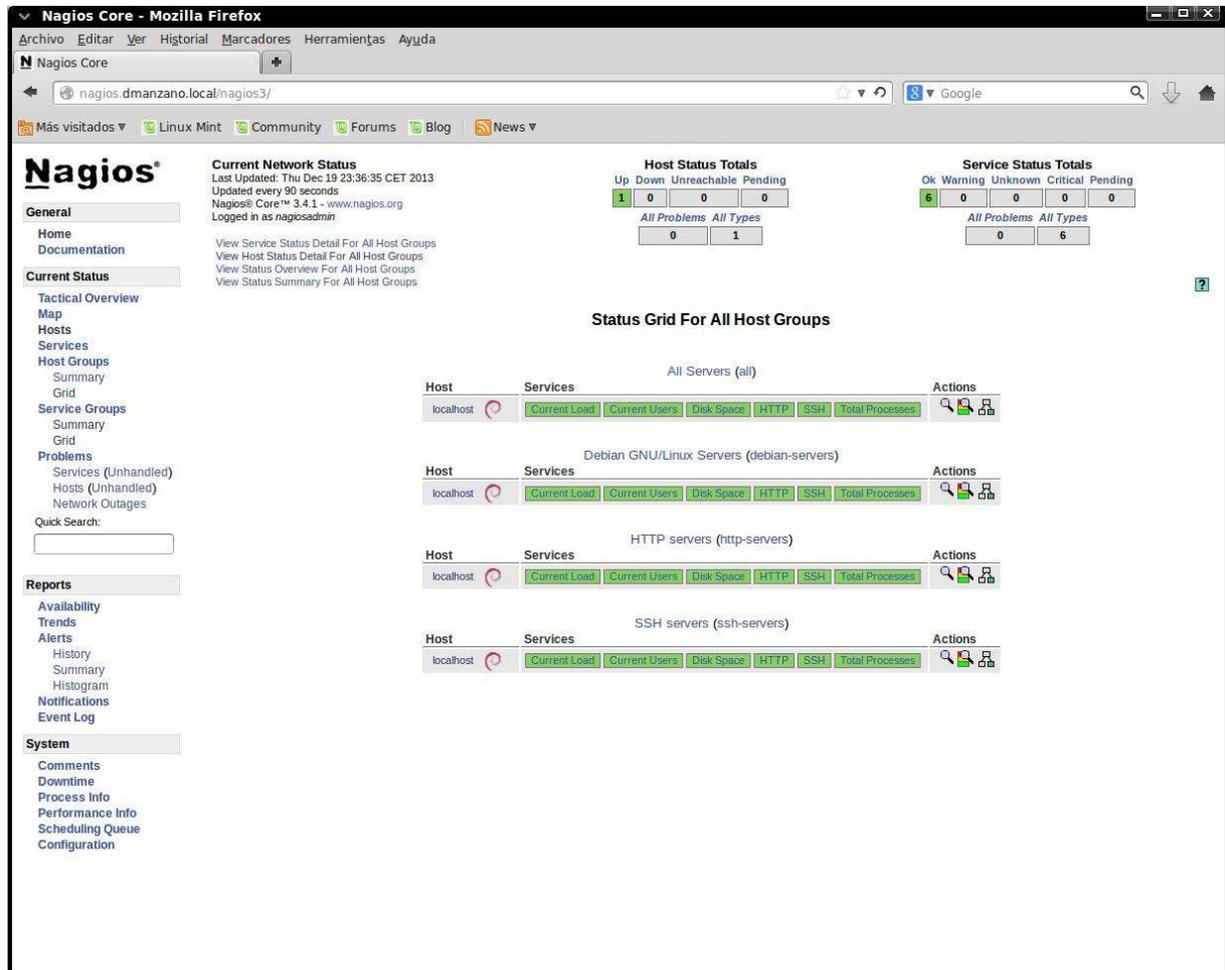
Nagios usa los estados soft y hard para describir el estado correcto de un host o un servicio. Si el estado es desconocido o diferente del estado anterior, se considera soft y Nagios lo comprueba varias veces para asegurarse de que es persistente. En ese momento pasa a hard.

Para nuestro ejemplo vamos a configurar el servidor Nagios, en el que aparecerá como monitorización su propia máquina, a través del servicio NRPE, y esta configuración será extensible al resto de servidores de la empresa y centralizará el control del estado de los mismos en una sola máquina.

Para ello nos vamos a la máquina virtual para Nagios que hemos creado y ejecutamos la instalación con un simple comando:

```
Se instalarán los siguientes paquetes NUEVOS:
apache2-mpm-prefork apache2-utils apache2.2-bin apache2.2-common autopoint
check-mk-config-nagios3 dbconfig-common dbus fontconfig fontconfig-config
fonts-droid fonts-liberation fping freeipmi-common freeipmi-tools gettext
ghostscript git git-man gosa gosa-plugin-nagios gosa-plugin-nagios-schema gsfnts
imagemagick-common javascript-common libaio1 libapache2-mod-php5 libapr1 libaprutil1
libaprutil1-dbd-sqlite3 libaprutil1-ldap libavahi-client3 libavahi-common-data
libavahi-common3 libcc-client2007e libcairo2 libcalendar-simple-perl
libclass-accessor-perl libclass-load-perl libclass-singleton-perl
libconfig-tiny-perl libcroco3 libcrypt-smhash-perl libcupsw2 libcupswimage2 libcurl3
libcurl3-gnutls libdata-oaf-list-perl libdate-manip-perl libdatetime-locale-perl
libdatetime-perl libdatetime-timezone-perl libdateutil libdbd-mysql-perl libdbi-perl
libdbus-1-3 libdigest-hmac-perl libdigest-md4-perl libencode-locale-perl
liberror-perl libffi5 libfile-listing-perl libfont-afm-perl libfontconfig1
libfpdf-tpl-php libfpdi-php libfreeipmi2 libgd-gd2-perl libgd2-xpm libgettextpo0
libglib2.0-0 libglib2.0-data libgomp1 libgs9 libgs9-common libhtml-form-perl
libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
libhtml-tree-perl libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl
libhttp-message-perl libhttp-negotiate-perl libice6 libijs-0.35 libio-pty-perl
libio-socket-ip-perl libio-socket-ssl-perl libipc-run-perl libipmiconsole2
libipmidetect0 libjansson4 libjasper1 libjbig0 libjbig2dec0 libjpeg8 libjs-jquery
libjs-jquery-ui libkohana2-php liblcms2-2 liblist-moreutils-perl liblqr-1-0 libltdl7
liblwp-mediatypes-perl liblwp-protocol-https-perl liblwp-useragent-determined-perl
libmagickcore5 libmagickwand5 libmail-imapclient-perl libmailtools-perl
libmath-calc-units-perl libmath-round-perl libmcrypt4 libmcmcacheid0
libmodule-implementation-perl libmodule-runtime-perl libmysqldclient18
libnagios-object-perl libnagios-plugin-perl libnet-daemon-perl libnet-dns-perl
libnet-http-perl libnet-ip-perl libnet-ldap-perl libnet-smtp-tls-perl
libnet-snmp-perl libnet-ssl-typer libodbc1 libonig2
libpackage-deprecationmanager-perl libpackage-stash-perl libpackage-stash-xs-perl
libpangl0-0 libpaper-utils libpaper libparams-classify-perl libparams-util-perl
libparams-validate-perl libparse-recdescent-perl libperl5.14 libppm-5.1-0
libpixman-1-0 libplrpc-perl libpng12-0 libpq5 libqdbm14 libradiusclient-ng2
libreadonly-perl libreadonly-xs-perl librecode0 librrd4 librrds-perl librtmp0
libruby1.9.1 librsensors4 libslp1 libsm6 libsnmp-base libsnmp15 libsocket-perl
libssh2-1 libsub-install-perl libsub-name-perl libsystemd-login0 libtaloc2 libtdb1
libthai-data libthai0 libtiff4 libtime-date-perl libtry-tiny-perl libunistring0
liburi-perl libwbclient0 libwww-perl libwww-robotrules-perl libxcb-render0
libxcb-shm0 libxft2 libxpm4 libxrender1 libxt6 libyaml-0-2 libyaml-syck-perl libzmq1
mlock mysql-client mysql-client-5.5 mysql-common mysql-server mysql-server-5.5
mysql-server-core-5.5 nagios-images nagios-nrpe-plugin nagios-nrpe-server
nagios-plugin-check-multi nagios-plugins nagios-plugins-basic nagios-plugins-common
nagios-plugins-contrib nagios-plugins-opensstack nagios-plugins-standard
nagios-snmp-plugins nagios3 nagios3-cgi nagios3-common nagios3-core nagios3-dbg
nagios3-doc nagiosgrapher ndoutils-common ndoutils-nagios3-mysql perlmagick php-fpdf
php5 php5-cli php5-common php5-curl php5-gd php5-imagick php5-imap php5-ldap
php5-mcrypt php5-mysql php5-recode pnp4nagios pnp4nagios-bin pnp4nagios-web
poppler-data python-httplib2 python-keystoneclient python-pkg-resources
python-prettytable gstat rrdtool rsync ruby ruby1.9.1 samba-common samba-common-bin
shared-mime-info slapd smartys3 smbclient snmp ssl-cert ttf-dejavu ttf-dejavu-core
ttf-dejavu-extra ttf-liberation uwsgi-core uwsgi-plugin-nagios wwwconfig-common
x11-common
0 actualizados, 258 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 116 MB de archivos.
Se utilizarán 436 MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

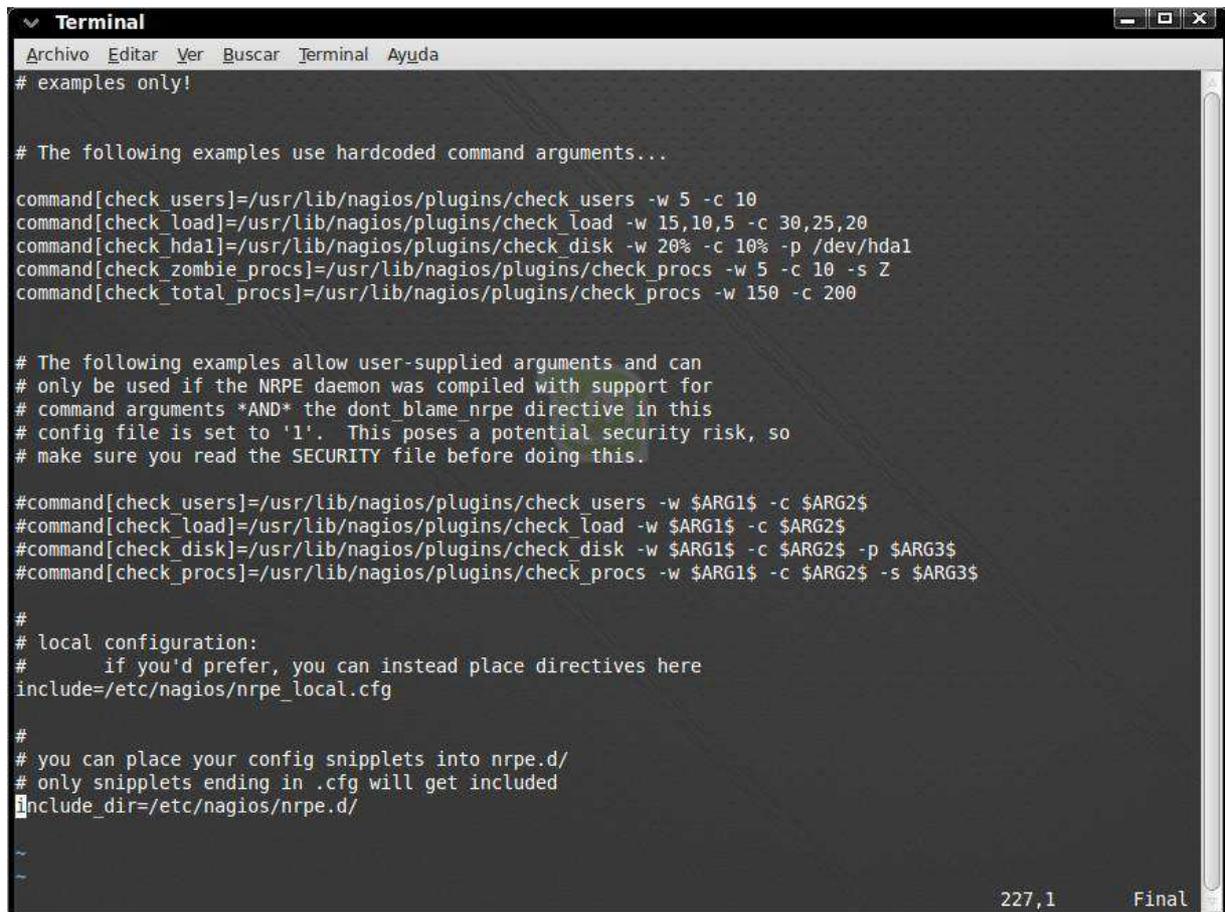
Ahora ya podemos acceder a la interfaz web de la máquina que ella misma nos ha configurado y levantado para acceder a nuestro servicio Nagios recién configurado y comprobar el estado del servidor en el que se encuentra instalado:



Como vemos estamos monitorizando, los usuarios que se encuentran conectados, la carga, el espacio de disco, el servicio web, el servicio SSH y el número total de procesos que se encuentran arrancados en dicha máquina.

Todo esto se encuentra configurado en el el siguiente archivo:

`/etc/nagios/nrpe.cfg`

A screenshot of a terminal window titled "Terminal" with a menu bar containing "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The terminal displays configuration examples for Nagios plugins. It starts with "# examples only!" and "# The following examples use hardcoded command arguments...". It lists four hardcoded commands: "command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10", "command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c 30,25,20", "command[check_hda1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /dev/hda1", and "command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s Z". It then shows user-supplied arguments with a warning: "# The following examples allow user-supplied arguments and can only be used if the NRPE daemon was compiled with support for command arguments *AND* the dont_blame_nrpe directive in this config file is set to '1'. This poses a potential security risk, so make sure you read the SECURITY file before doing this." It lists four user-supplied commands: "#command[check_users]=/usr/lib/nagios/plugins/check_users -w \$ARG1\$ -c \$ARG2\$", "#command[check_load]=/usr/lib/nagios/plugins/check_load -w \$ARG1\$ -c \$ARG2\$", "#command[check_disk]=/usr/lib/nagios/plugins/check_disk -w \$ARG1\$ -c \$ARG2\$ -p \$ARG3\$", and "#command[check_procs]=/usr/lib/nagios/plugins/check_procs -w \$ARG1\$ -c \$ARG2\$ -s \$ARG3\$". It also shows local configuration: "# local configuration:", "# if you'd prefer, you can instead place directives here", "include=/etc/nagios/nrpe_local.cfg", "#", "# you can place your config snippets into nrpe.d/", "# only snippets ending in .cfg will get included", and "include_dir=/etc/nagios/nrpe.d/". The terminal ends with "~" and "~" on separate lines. The bottom right corner shows "227,1" and "Final".

Gracias este sistema, y mediante la instalación de plugins podremos monitorizar cualquier elemento de la máquina, un listado de los diferentes plugins existentes para nagios lo podemos encontrar aquí:

<http://exchange.nagios.org/directory/Plugins>

Y de ahí podemos descargarlos de forma completamente gratuita para implementarlos en nuestro sistema e incluso podremos modificarlos o crear nuestros propios plugins y subirlos para colaborar con la comunidad.

Utilizando este sistema garantizamos un control sobre toda nuestra infraestructura, ya que nagios se puede configurar para emitir alertas y correos sobre la posible falla de cualquiera de los servicios, incluso acciones automáticas que se puede programar para que nagios las realice en caso de que se produzca un evento concreto.

3.4 Control de usuarios

En nuestra empresa vamos a necesitar centralizar la gestión de usuarios, es por ello que vamos a implementar un sistema de directorio de usuarios que nos permita configurar mucho del software que se está desplegando en la empresa de forma que exista un único punto de control. Si no instalásemos un sistema de gestión de usuarios como este tendríamos que estar añadiendo uno por uno y dando de alta a los usuarios en los diferentes servicios, como puede ser la mensajería o el correo electrónico.

Frente a la extendida alternativa de pago Active Directory de Microsoft nosotros vamos a implementar el sistema OpenLDAP que nos va a permitir disponer de una funcionalidad muy parecida sin necesidad de realizar pagos de licencias.

3.4.1 OpenLDAP



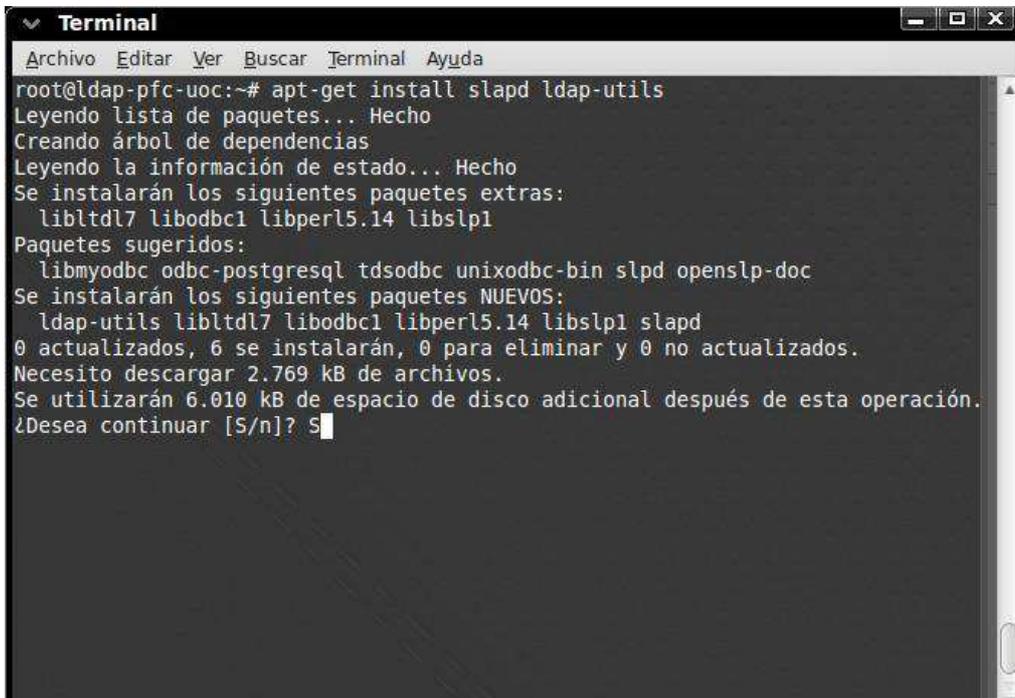
Un servidor LDAP es un servidor de datos orientado al almacenamiento de datos de usuarios a modo de directorio y optimizado para la realización rápida de consultas de lectura, tiene como principal utilidad el funcionar como servidor de autenticación para los distintos servicios de un sistema informático como puedan ser: entrar en un PC, acceder a un servidor ftp, entrar en una aplicación web, acceder a servidores de correo entrante POP3 y saliente SMTP, etc... No es más que un gestor de bases de datos jerárquico, pero además tiene un diseño optimizado para realizar cientos de consultas simultáneas de forma eficiente, puesto que es la operación más común.

Al configurar todos los PCs y todos los servicios de la red para que se autenticuen en un servidor LDAP del que disponemos en nuestra red, basta con crear cuentas de usuario y grupos de usuarios en dicho servidor para que éstos puedan hacer uso del sistema y de sus servicios desde cualquier puesto de la red. Así se centraliza la administración de usuarios en un único lugar.

OpenLDAP posee tres componentes principales:

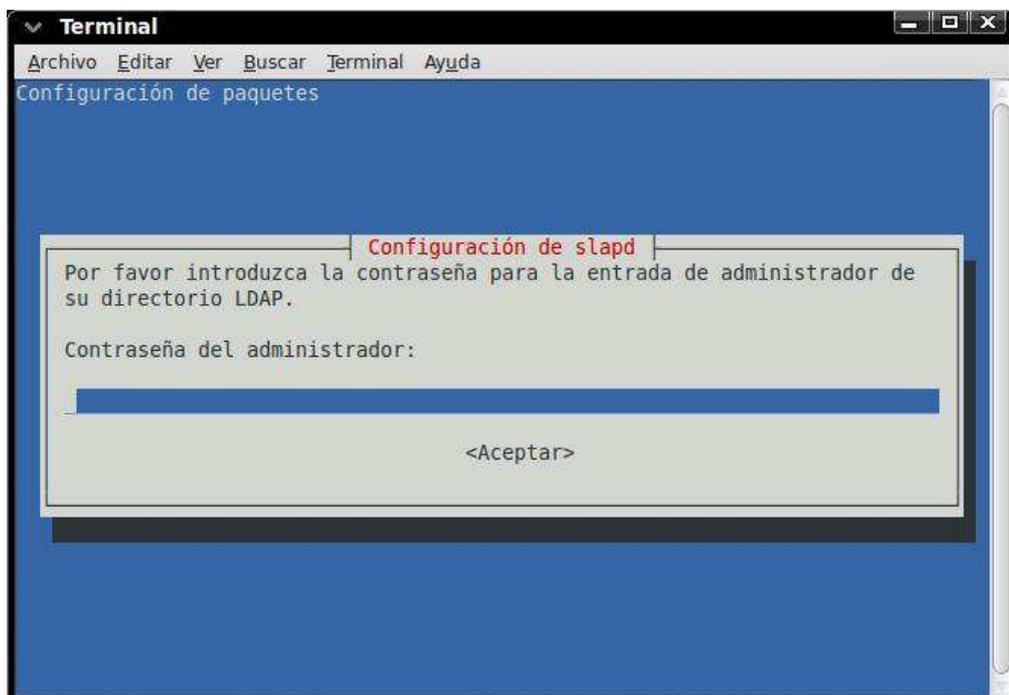
- slapd -Dominio de servidor y herramientas
- Bibliotecas que implementan el protocolo LDAP
- Programas cliente: ldapsearch, ldapdelete...

Para la instalación de OpenLDAP en nuestro sistema volvemos hacerlo mediante el comando apt-get de Debian, que incluye en sus repositorios una versión de OpenLDAP:



```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
root@ldap-pfc-uoc:~# apt-get install slapd ldap-utils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 libltdl7 libodbc1 libperl5.14 libsasl1
Paquetes sugeridos:
 libmyodbc odbc-postgresql tdsodbc unixodbc-bin slpd openslp-doc
Se instalarán los siguientes paquetes NUEVOS:
 ldap-utils libltdl7 libodbc1 libperl5.14 libsasl1 slapd
0 actualizados, 6 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 2.769 kB de archivos.
Se utilizarán 6.010 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? S
```

Dejamos que el sistema descargue todo lo que necesita y nos preguntará por la clave de administrador del directorio de usuarios que estamos configurando:



```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Configuración de paquetes

Configuración de slapd
Por favor introduzca la contraseña para la entrada de administrador de
su directorio LDAP.

Contraseña del administrador:
[Redacted password]

<Aceptar>
```

Una vez que termine la instalación nuestro sistema arrancará automáticamente el demonio del nuevo servicio, y podremos ejecutar un comando para ver que todo se ha creado correctamente:

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
e.)
Desempaquetando libltdl7:amd64 (de .../libltdl7_2.4.2-1.1_amd64.deb) ...
Seleccionando el paquete libodbc1:amd64 previamente no seleccionado.
Desempaquetando libodbc1:amd64 (de .../libodbc1_2.2.14p2-5_amd64.deb) ...
Seleccionando el paquete libperl5.14 previamente no seleccionado.
Desempaquetando libperl5.14 (de .../libperl5.14_5.14.2-21+deb7u1_amd64.deb) ...
Seleccionando el paquete libslp1 previamente no seleccionado.
Desempaquetando libslp1 (de .../libslp1_1.2.1-9_amd64.deb) ...
Seleccionando el paquete slapd previamente no seleccionado.
Desempaquetando slapd (de .../slapd_2.4.31-1+nmu2_amd64.deb) ...
Seleccionando el paquete ldap-utils previamente no seleccionado.
Desempaquetando ldap-utils (de .../ldap-utils_2.4.31-1+nmu2_amd64.deb) ...
Procesando disparadores para man-db ...
Configurando libltdl7:amd64 (2.4.2-1.1) ...
Configurando libodbc1:amd64 (2.2.14p2-5) ...
Configurando libperl5.14 (5.14.2-21+deb7u1) ...
Configurando libslp1 (1.2.1-9) ...
Configurando slapd (2.4.31-1+nmu2) ...
  Creating new user openldap... done.
  Creating initial configuration... done.
  Creating LDAP directory... done.
[ ok ] Starting OpenLDAP: slapd.
Configurando ldap-utils (2.4.31-1+nmu2) ...
root@ldap-pfc-uoc:~#
```

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
Configurando libslp1 (1.2.1-9) ...
Configurando slapd (2.4.31-1+nmu2) ...
  Creating new user openldap... done.
  Creating initial configuration... done.
  Creating LDAP directory... done.
[ ok ] Starting OpenLDAP: slapd.
Configurando ldap-utils (2.4.31-1+nmu2) ...
root@ldap-pfc-uoc:~# slapcat
dn: dc=dmanzano,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: dmanzano.local
dc: dmanzano
structuralObjectClass: organization
entryUUID: 7108adae-fe71-1032-9bc8-4989f08bc3e9
creatorsName: cn=admin,dc=dmanzano,dc=local
createTimestamp: 20131221095309Z
entryCSN: 20131221095309.168643Z#000000#000#000000
modifiersName: cn=admin,dc=dmanzano,dc=local
modifyTimestamp: 20131221095309Z

dn: cn=admin,dc=dmanzano,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: eINTSEF9cXIwUktXam44WU4yY21vT25YaVFacnBpRU1WSlFTWEs=
structuralObjectClass: organizationalRole
entryUUID: 710957f4-fe71-1032-9bc9-4989f08bc3e9
creatorsName: cn=admin,dc=dmanzano,dc=local
createTimestamp: 20131221095309Z
entryCSN: 20131221095309.173006Z#000000#000#000000
modifiersName: cn=admin,dc=dmanzano,dc=local
modifyTimestamp: 20131221095309Z
root@ldap-pfc-uoc:~#
```

Aquí vemos como hemos creado el dominio `dmanzano.local` y el usuario `admin` perteneciente a dicho dominio, que será el que nos permita autenticarnos contra el directorio para configurarlo y crear nuevos usuarios. Así nuestro usuario administrador quedará de la siguiente forma:

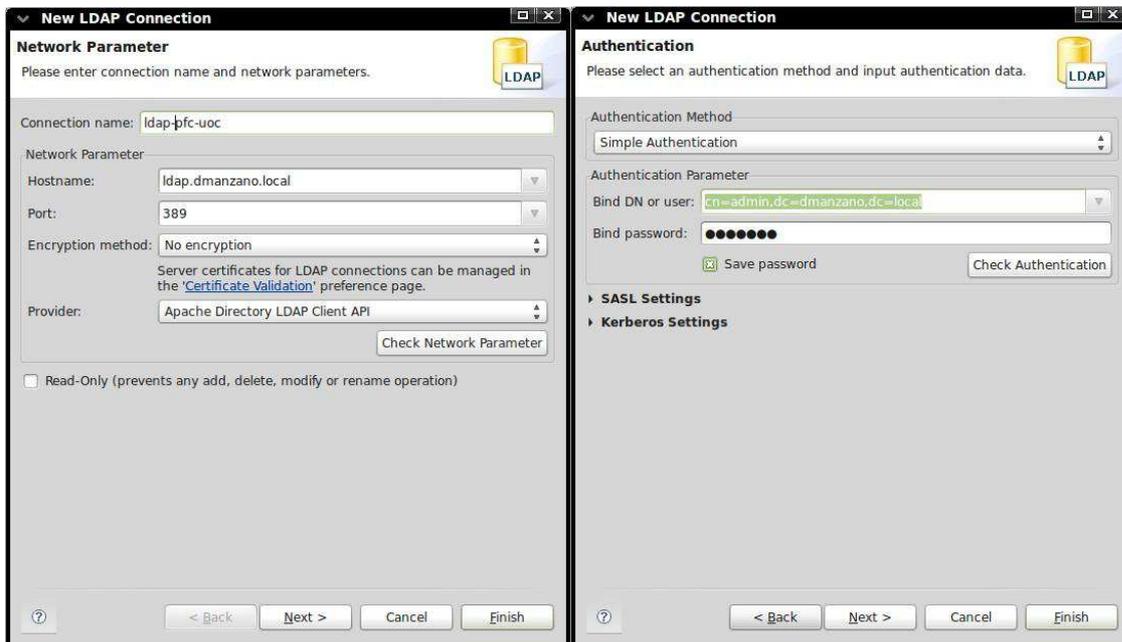
`cn=admin, dc=dmanzano, dc=local`



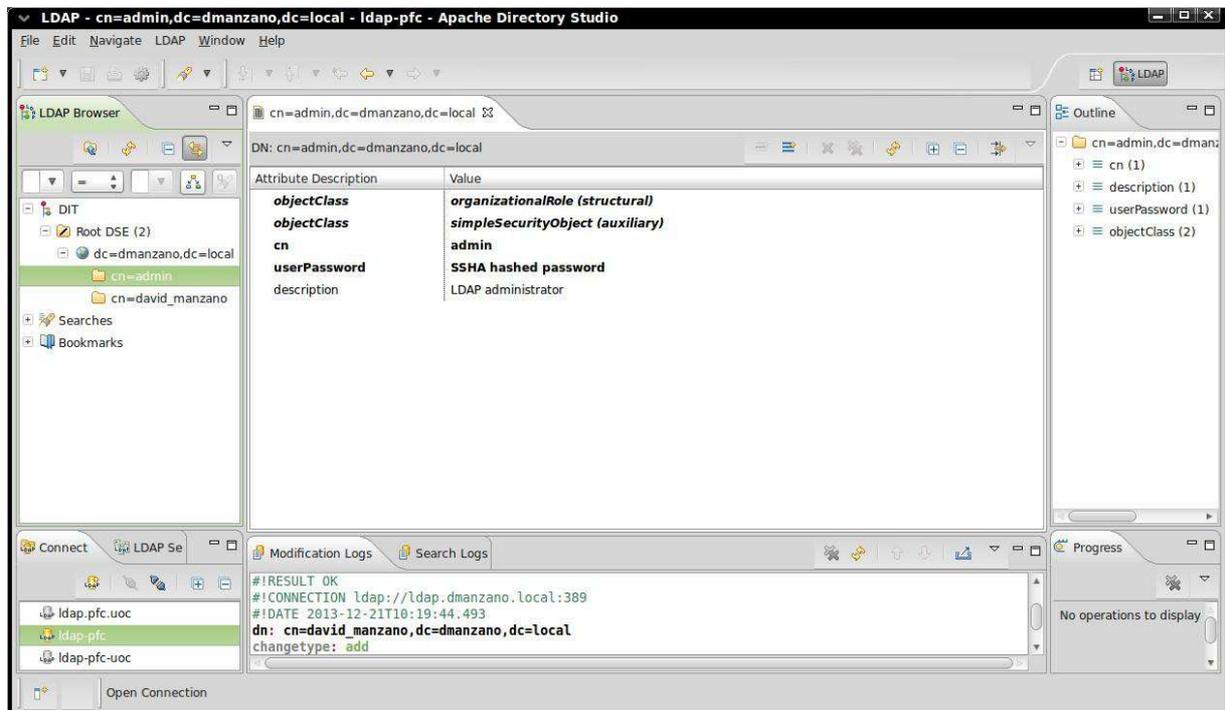
Ahora para facilitarnos el manejo de dicho directorio volveremos a recurrir a las posibilidades que el software libre nos ofrece, y nos descargaremos el programa Apache Directory Studio:

<http://directory.apache.org/studio/downloads.html>

Y tras instalarlo en nuestro sistema debían lo ejecutaremos para conectar contra el directorio de usuarios que acabamos de crear:



Una vez conectados veremos la rama de usuarios básica con el usuario administrador y ya podemos configurar todos los usuarios como queramos:



En la imagen anterior vemos cómo hemos generado un nuevo usuario `david_manzano` dentro de la rama global. A partir de este momento todos los sistemas cuya autenticación configuremos contra el directorio LDAP bajo esta rama, contarán con el usuario `david_manzano` y éste podrá acceder al mismo bajo los permisos especificados para dicha rama en el programa configurado.

De este modo centralizamos toda la gestión de usuario de la empresa bajo un mismo sistema y no tendremos que andar duplicando usuarios por todos los sistemas de la empresa. Cada empleado tiene un usuario único en este directorio con una clave, y ese usuario y clave le permite autenticarse en todos los servicios de la empresa, correo, mensajería, subversion, etc.

3.5 Comunicaciones

3.5.1 Gestión del correo: Zimbra

Para la gestión integral del sistema de correo electrónico de la empresa se ha elegido el paquete completo Zimbra, esta elección la hemos realizado teniendo en cuenta diversos factores, pero el más importante ha sido la garantía de estabilidad y solidez que ofrece el sistema.

Al contar detrás con una empresa como es VMware, esto garantiza que dispondremos de un soporte que puede ser clave en una de los elementos más importantes dentro de la empresa.

Zimbra es una solución completa de correo electrónico, libreta de direcciones, calendario y tareas compatible con clientes propietarios tales como Microsoft Outlook, Novell Evolution y Apple Mail y también provee soporte de sincronización nativo de dos vías para muchos dispositivos móviles.

Ventajas de Zimbra

- calendarios compartidos, movilidad o gestión de documentos.
- Coste mucho más bajo.
- No hay dependencia de un proveedor.
- Puedes utilizar el cliente web Zimbra junto con el resto de clientes tradicionales.
- Tecnología de Plugins que permite integrar aplicaciones y módulos en Zimbra

Módulo de correo

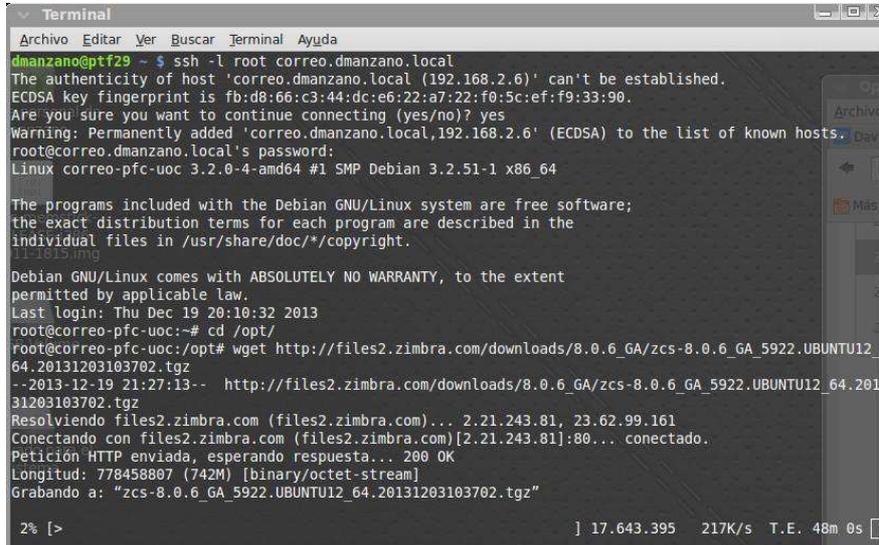
Para gestionar el correo existe un módulo completo, sencillo de utilizar, con las siguientes características:

- Vista por conversación: Zimbra busca y muestra por conversación todos los correos.
- Compartir carpetas de correo, con otros compañeros.
- Suscripción a canales corporativos de noticias o externos.
- Opción de guardar nuestras búsquedas más habituales.
- Agregar cuentas externas POP3/IMAP
- Tecnología "Arrastrar y soltar"

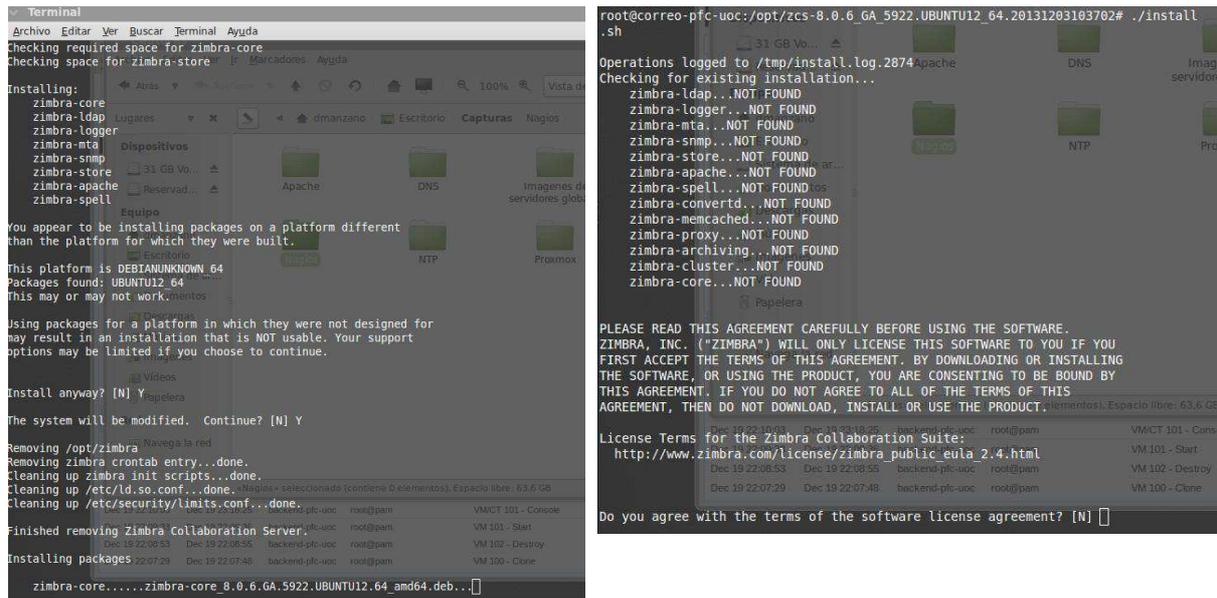
Zimbra mantiene el espíritu de Open Source en su versión "community", la cual es posible descargar directamente desde su página web, junto con el código fuente de la misma.

Para la implementación del mismo dentro de nuestro sistema, hemos creado una máquina virtual en nuestro DMZ que será la encargada de alojar dicho servicio. La ubicación de esta máquina en el DMZ, no se considera una solución ideal, puesto que la máquina de correo contiene información muy sensible de los usuarios como es el correo electrónico, pero en nuestro caso por simplificar el sistema, lo implementaremos de esta manera. La solución óptima hubiese sido implementar un gateway de correo, como puede ser el *F1 mail Gateway* que haría de intermediario entre el exterior y nuestro sistema de correo alojado en el backend, pero puesto que por falta de tiempo este sistema no se va a poder implementar se ha tomado la opción de implementar el correo en el DMZ como una alternativa más segura a la de permitir acceso desde el exterior al backend de nuestra infraestructura.

Para dicha implementación lo primero será descargarnos el paquete de Zimbra en nuestro sistema:



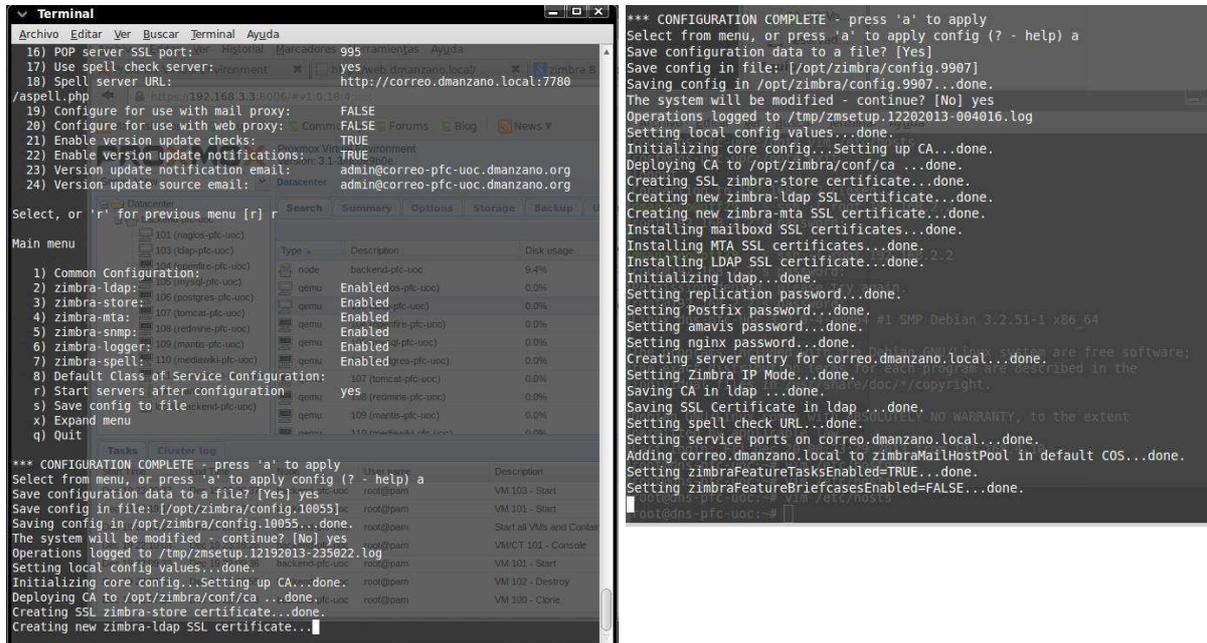
Y una vez descargado comenzamos la instalación y el despliegue del mismo:



Cuando hemos terminado la instalación nos dará acceso a un menú básico que nos permitirá la configuración del mismo:

Implementación de GNU/Linux para una empresa de desarrollo de software

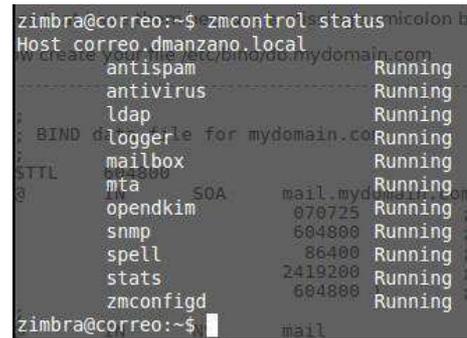
David Manzano Fiestas



```
Terminal
16) POP server SSL port: 995
17) Use spell check server: Yes
18) Spell server URL: http://correo.dmanzano.local:7780
19) Configure for use with mail proxy: FALSE
20) Configure for use with web proxy: FALSE
21) Enable version update checks: TRUE
22) Enable version update notifications: TRUE
23) Version update notification email: admin@correo-pfc-uoc.dmanzano.org
24) Version update source email: admin@correo-pfc-uoc.dmanzano.org

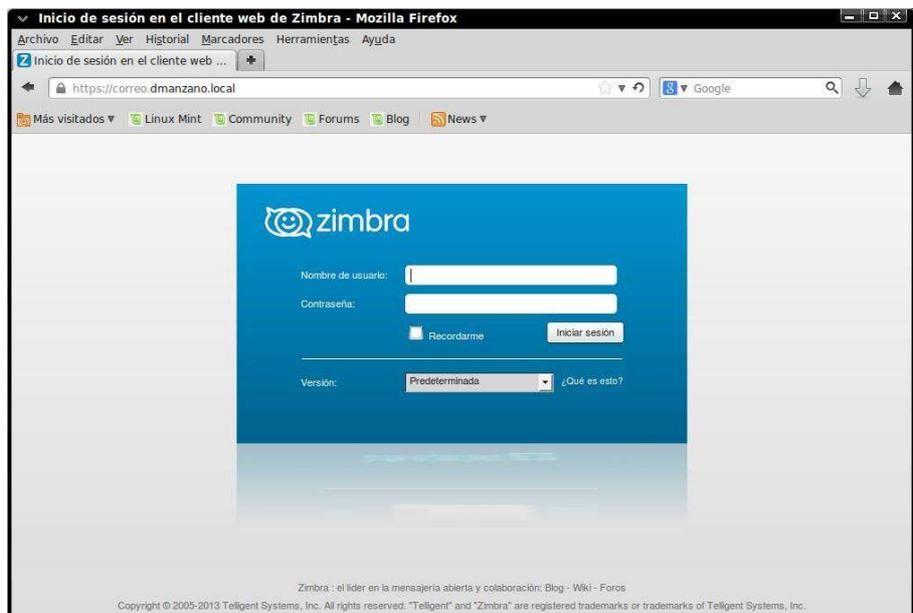
*** CONFIGURATION COMPLETE -- press 'a' to apply
Select from menu, or press 'a' to apply config (? - help) a
Save configuration data to a file? [Yes] yes
Save config in file: [/opt/zimbra/config.10055]
Saving config in /opt/zimbra/config.10055...done.
The system will be modified - continue? [No] yes
Operations logged to /tmp/zmsetup.12192013-235022.log
Setting local config values...done.
Initializing core config...Setting up CA...done.
Deploying CA to /opt/zimbra/conf/ca ...done.
Creating SSL zimbra-store certificate...done.
Creating new zimbra-ldap SSL certificate...done.
Creating new zimbra-mta SSL certificate...done.
Installing mailboxd SSL certificates...done.
Installing MTA SSL certificates...done.
Installing LDAP SSL certificate...done.
Initializing ldap...done.
Setting replication password...done.
Setting Postfix password...done.
Setting amavis password...done.
Setting nginx password...done.
Creating server entry for correo.dmanzano.local...done.
Setting Zimbra IP Mode...done.
Saving CA in ldap...done.
Saving SSL Certificate in ldap...done.
Setting spell check URL...done.
Setting service ports on correo.dmanzano.local...done.
Adding correo.dmanzano.local to zimbraMailHostPool in default COS...done.
Setting zimbraFeatureTasksEnabled=TRUE...done.
Setting zimbraFeatureBriefcasesEnabled=FALSE...done.
```

Y ahora entramos con el usuario Zimbra en el sistema y comprobamos que todos los servicios de Zimbra están funcionando adecuadamente:



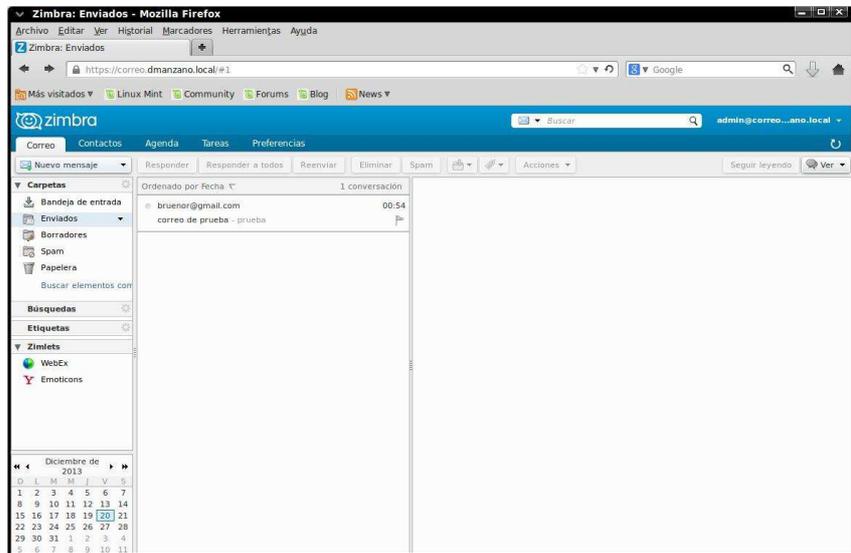
```
zimbra@correo:~$ zmcontrol status
Host correo.dmanzano.local
-----
antispam Running
antivirus Running
ldap Running
logger Running
mailbox Running
mta Running
opendkim Running
snmp Running
spell Running
stats Running
zmconfigd Running
zimbra@correo:~$
```

Entonces estamos listos para acceder vía web al nuevo servidor de correo que acabamos de configurar:



Y dentro podremos ver nuestra nueva bandeja de entrada y hacemos una prueba de envío y recepción de correo:

Gracias a Zimbra solventamos una de las mayores necesidades de cualquier empresa, a la vez que añadimos algunas opciones que aportan un valor añadido como es una agenda y calendarios colaborativos o una lista de tareas que también pueden ser compartidas por los diversos usuarios de la empresa.



Aunque la versión Community de Zimbra no soporta la realización de copias de seguridad, al haber implementado el sistema en una máquina virtual de Proxmox, tenemos esta funcionalidad cubierta gracias el sistema propio de éste, así que garantizamos que los correos de todos nuestros empleados de la empresa estarán perfectamente a salvo en caso de que se produzca algún tipo de desastre.

En nuestro caso, hemos optado por no emplear el sistema POP, puesto que esto implicaría no mantener los mensajes de lo usuarios en el servidor, y habilitar únicamente IMAP y SMTP para envío y recepción.

Por último remarcamos lo importante que será el habilitar el servicio SSL para éste de forma que todas las conexiones web sean seguras y que nadie pueda capturar la clave de nuestros usuarios cuando se autenticuen desde redes inseguras.

Aunque Zimbra solo garantiza la compatibilidad con distribuciones Ubuntu y no con Debian, hemos utilizado el mismo paquete para implementar dicho sistema en nuestro servidor, y aunque ha habido que realizar algunas configuraciones e instalación de librerías específicas y especiales por estar en una distribución que no es la oficialmente soportada, el despliegue del mismo se ha conseguido con éxito.

3.5.2 Mensajería: Openfire

Openfire es un sistema de mensajería instantánea GPL y hecho en java que utiliza el protocolo XMPP.

Es un servidor de colaboración en tiempo real (RTC) bajo la licencia de código abierto Apache. Es increíblemente fácil de configurar y administrar, pero ofrece seguridad y rendimiento sólidos.

Openfire además implementa las siguientes características que hacen que sea una solución muy versátil y potente que se podrá integrar en nuestra empresa de diversas formas y generar un entorno colaborativo muy eficiente :

- Panel de administración web
- Interfaz para agregar plugins
- Soporte SSL/TLS
- Guía de instalación e interfaz web amigables.
- Adaptable según las necesidades
- Conferencias
- Interacción con MSN, Google Talk, Yahoo messenger, AIM, ICQ, Jingle
- Cluster con múltiples servidores
- Transferencia de Archivos
- Compresión de datos
- Tarjetas personales con Avatar
- Mensajes offline
- Favoritos
- Estadísticas del Servidor, mensajes, paquetes, etc.
- Autenticación vía Certificados, Kerberos, LDAP, PAM y Radius
- Almacenamiento en Active Directory, LDAP, MS SQL, MySQL, Oracle y PostgreSQL
- SASL: ANONYMOUS, DIGEST-MD5 y Plain

Como vemos Openfire es una herramienta que va un poco más allá que los sistemas tradicionales de mensajería. Es por ello que una vez instalada contaremos con una interfaz web que os permitirá configurar todos estos servicios, pero hasta llegar ahí seguimos todo el proceso que se describe a continuación:

Para empezar descargamos el paquete oficial de Openfire de los repositorios de Ignite:

<http://www.igniterealtime.org/downloads/>

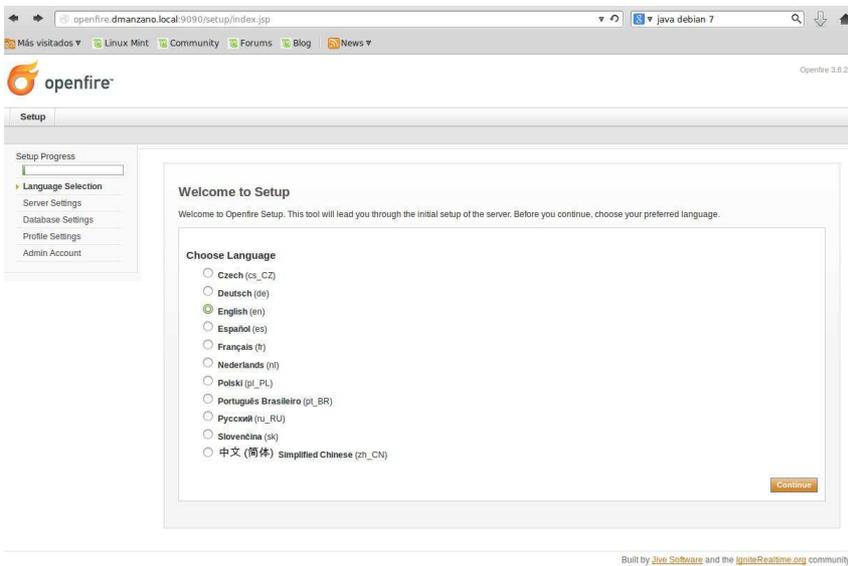
Y una vez en nuestro sistema lo descomprimiremos. Como el sistema utiliza Java, vamos a copiar a nuestro sistema el paquete descargado de la web de oracle que también deberemos instalar y sin el cual Openfire no funcionaría:

```
dmanzano@ptf29 ~/Descargas $ scp oracle-j2sdk1.7_1.7.0+update45_amd64.deb root@openfire:/opt
The authenticity of host 'openfire (192.168.3.7)' can't be established.
ECDSA key fingerprint is c7:9b:32:31:fe:82:d2:29:35:09:28:29:04:23:9b:4b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'openfire' (ECDSA) to the list of known hosts.
root@openfire's password:
oracle-j2sdk1.7_1.7.0+update45_amd64.deb 100% 131MB 11.0MB/s 00:12
dmanzano@ptf29 ~/Descargas $
```

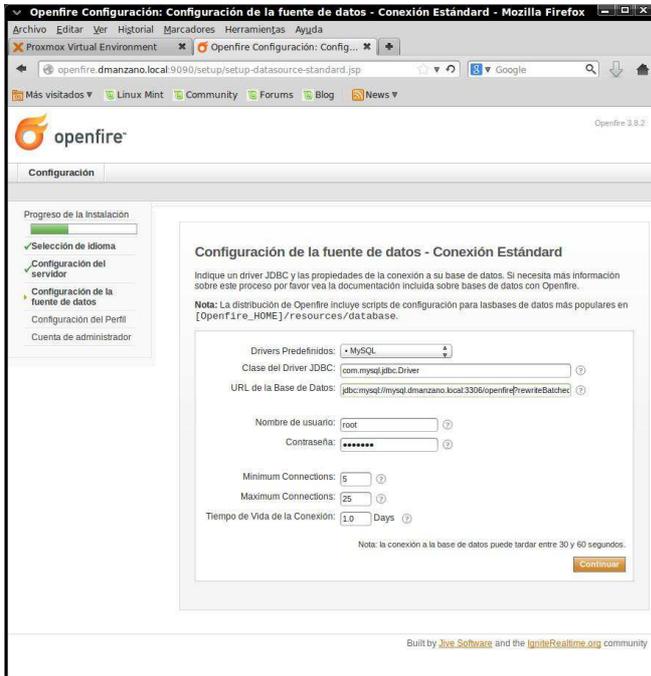
Y una vez descomprimido, accedemos al directorio bin del mismo y ejecutamos el comando que iniciará nuestro servidor:

```
root@openfire-pfc-uoc:/opt/openfire/bin# ./openfire start
testing JVM in /usr ...
Starting openfire
root@openfire-pfc-uoc:/opt/openfire/bin# nohup: se añade la salida a «nohup.out»
```

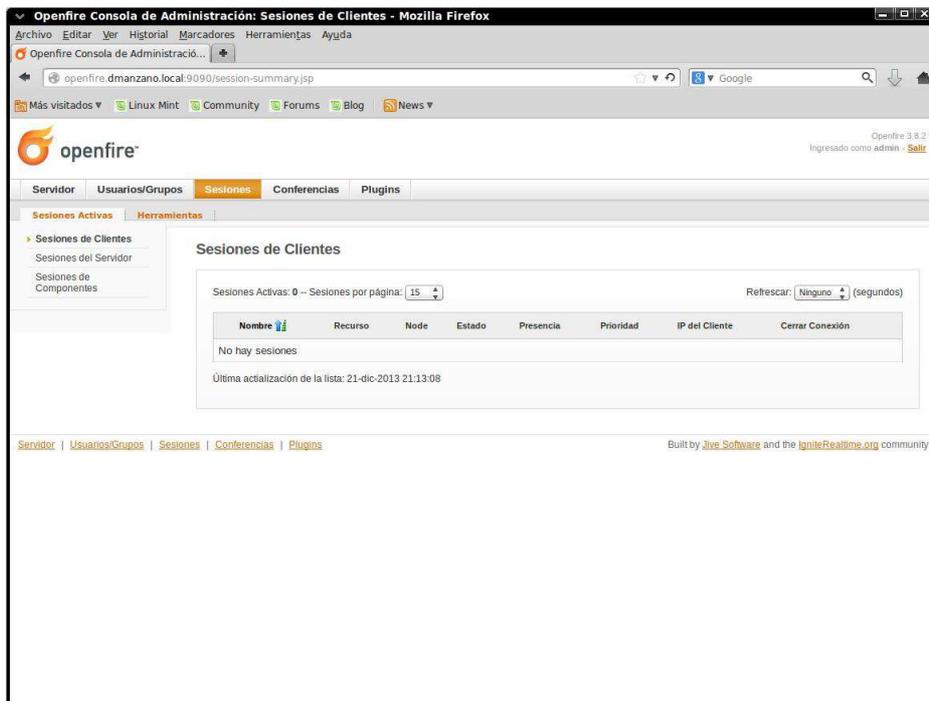
Ahora ya podemos acceder al puerto 9090 de nuestro servidor que nos servirá para configurar nuestro sistema openfire:



Este wizard solo aparecerá la primera vez que iniciamos el sistema, y para configurarlo vamos a utilizar algunos de las herramientas de las que ya disponemos en nuestra red local como son el servidor de Mysql o el directorio de usuarios LDAP:



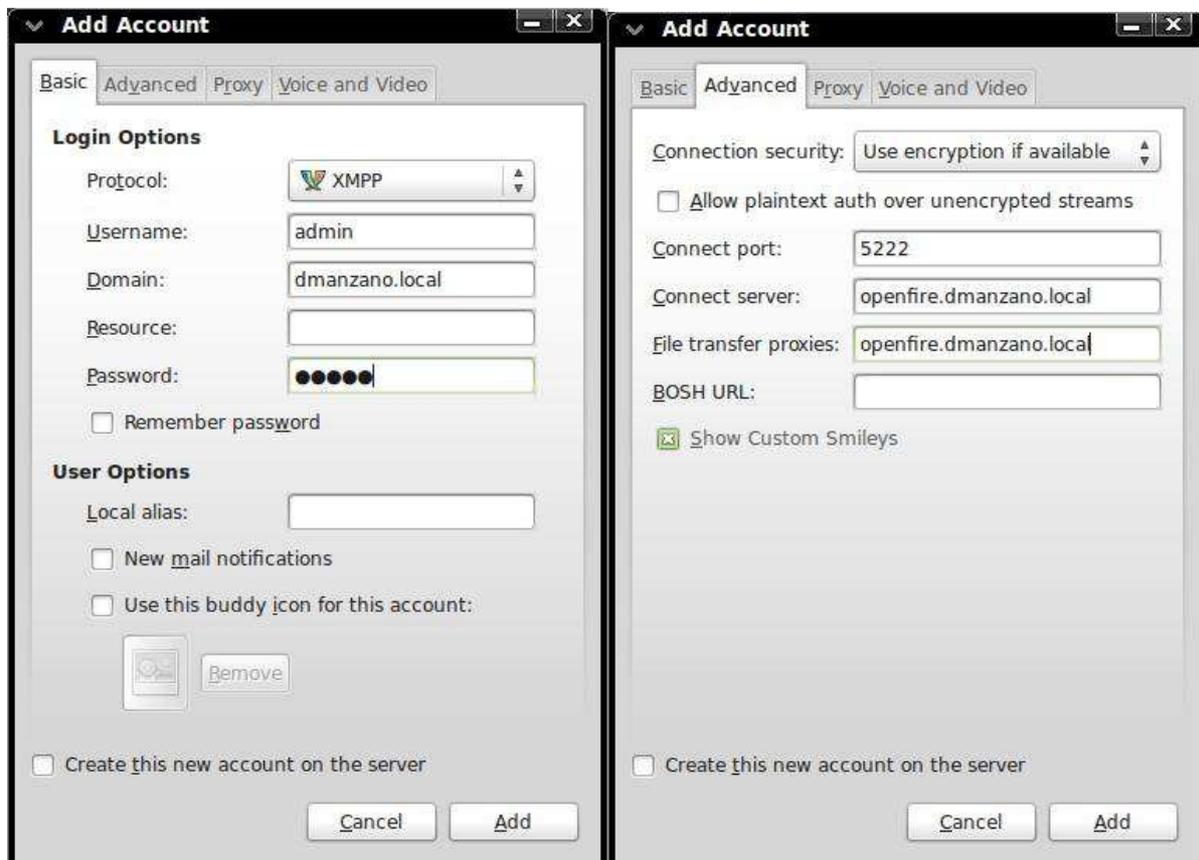
En esta imagen vemos como Openfire se integra con la máquina de nuestra red que ya contiene el servidor de base de datos Mysql corporativo. Hasta que por fin conseguimos alcanzar el final de la configuración.



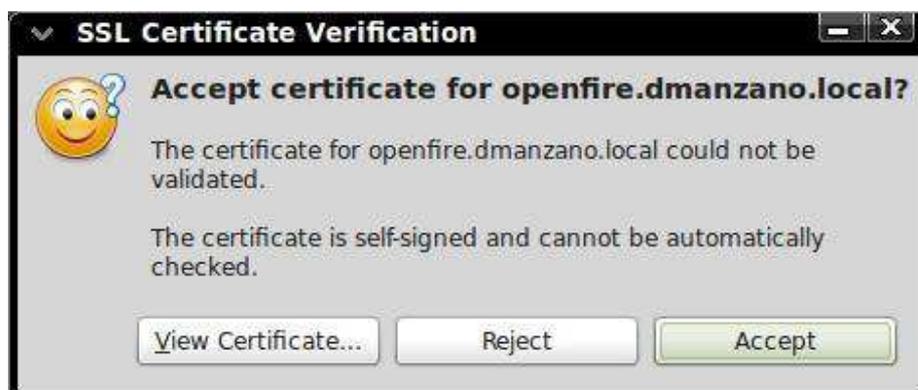
Ahora que ya tenemos configurado nuestro sistema openfire es el momento de configurar nuestro cliente de mensajería instantánea y comprobar que conecta con el servidor.

Como cliente de mensajería instantánea hemos escogido Pidgin, probablemente por ser uno de los más estables y conocidos que soportan el protocolo XMPP.

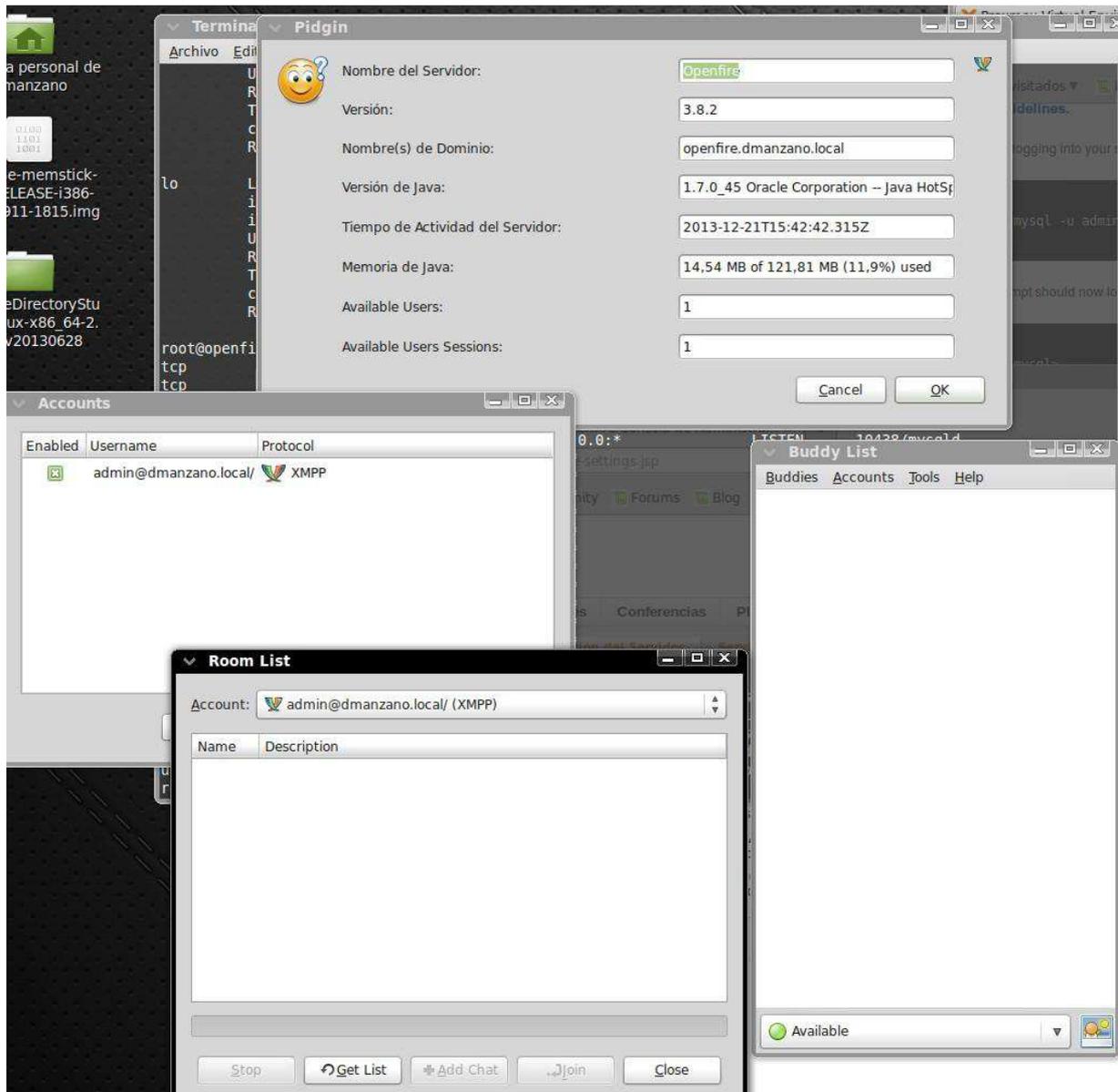
Agregamos la cuenta en nuestro servidor:



Y vemos como nos pide confirmar el certificado de seguridad utilizado para la transmisión segura:



Y ya estamos conectados a nuestro propio servidor de mensajería instantánea:



Siempre podemos tomar la opción de publicar el servidor de mensajería al exterior permitiendo que usuarios de otras empresas o externos se conecten para interactuar con el personal de la empresa, pero esto quedará a elección de la dirección de IT de la empresa en función de las necesidades estratégicas, por lo pronto nosotros hemos considerado esto como un servicio meramente interno y para uso exclusivo desde la intranet de la empresa.

3.5.3 VPN

Una red privada virtual (VPN) extiende una red privada a través de una red pública, como Internet.

Permite a un equipo enviar y recibir datos a través de redes compartidas o públicas como si estuviera conectado directamente a la red privada, mientras se benefician de las políticas de funcionalidad, seguridad y gestión de la red privada. Esto se hace mediante la creación de una conexión virtual punto a punto mediante el uso de conexiones dedicadas, cifrado, o una combinación de ambos.

Las VPN permiten a los empleados acceder de forma segura a la intranet de su empresa mientras estén de viaje fuera de la oficina. De forma similar, las VPN conectan de forma segura y rentable oficinas de una misma empresa que están alejadas, creando una red virtual cohesionada.

Las VPN pueden ser de acceso remoto (conexión de un equipo individual a una red) o de sitio a sitio (que conecta dos redes entre sí). Una VPN también puede ser utilizado para interconectar dos redes similares través de una red medio diferente. Por ejemplo, dos redes de IPv6 en una red IPv4.

Sistemas VPN se pueden clasificar por:

- los protocolos que se utilizan para hacer un “túnel” al tráfico de datos.
- la ubicación del punto de fin de túnel, por ejemplo, en el extremo de la red del cliente o proveedor.
- si ofrecen conectividad de sitio a sitio o de acceso remoto.
- los niveles de seguridad proporcionado.
- la capa OSI que presentan a la red de conexión, tales como Layer 2 circuitos o conectividad de Capa 3 de red.

Para impedir la divulgación de información privada, las VPN suelen permitir sólo el acceso remoto autenticado y hacen uso de técnicas de encriptación y de protocolos tunelizados.

El modelo de seguridad de VPN proporciona:

- confidencialidad de tal manera que un atacante sólo vería los datos cifrados
- autenticación del remitente para evitar que usuarios no autorizados accedan a la VPN.
- integridad de los mensajes para detectar cualquier caso de manipulación de los mensajes transmitidos

Los protocolos VPN seguras incluyen lo siguiente:

- Internet Protocol Security (IPsec). Su diseño se ajusta a la mayoría de los objetivos de seguridad: autenticación, integridad y confidencialidad. IPsec usa encriptación, encapsulando un paquete IP dentro

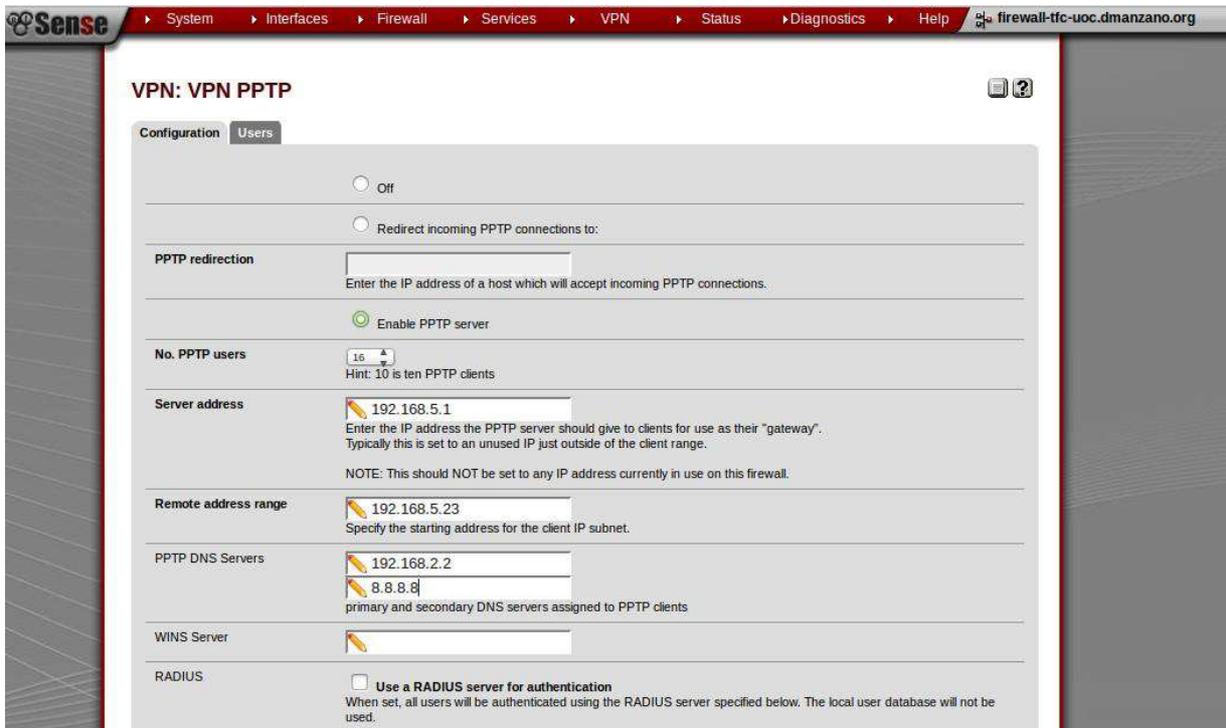
de un paquete IPsec. El desencapsulado tiene lugar al final del túnel, donde el paquete IP original es descifrado y devuelto a su destino previsto.

- Transport Layer Security (SSL/TLS) puede tunelar todo el tráfico de una red o asegurar una conexión individual.
- Datagram Transport Layer Security (DTLS) - usado en Cisco AnyConnect VPN y en OpenConnect VPN para resolver los problemas que SSL /TLS tiene con túneles a través de UDP.
- Microsoft Point-to-Point Encryption (MPPE) trabaja con los protocolos de tunelación Punto a Punto y en varias implementaciones compatibles en otras plataformas.
- Microsoft Secure Socket Tunneling Protocol (SSTP) trabaja con Protocolos de túneles Punto a Punto (PPP)
- Multi Path Virtual Private Network (MPVPN). Ragula Systems Development Companyes propietaria de la marca registrada " MPVPN ".
- VPN-PPTP que se define como una VPN punto a punto que permite que clientes se conecten a nuestro servidor estableciendo un canal de comunicación seguro.

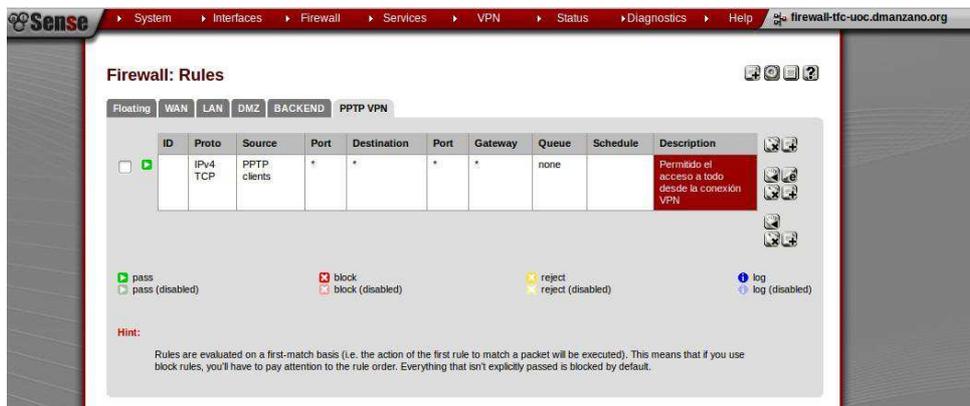
En nuestro caso, vamos a configurar un servidor VPN gracias al software incorporado en nuestro Firewall. Pfsense nos va a dejar configurar diversos tipos de VPN, que permitirá que los desarrolladores trabajen desde fuera de la empresa como si estuvieran físicamente conectados a la misma.

Aunque Pfsense nos va a permitir diversos tipos de VPN hemos escogido por comodidad implementar PPTP siendo esta la más fácil de configurar puesto que se basa simplemente en un usuario y una contraseña que deberemos dar a nuestros empleados para que se autenticuen con nuestra Red. En la práctica este sistema no sería el escogido para la implementación, puesto que el cifrado de VPN-PPTP ya no es lo suficientemente fuerte como para considerarse seguro, por lo que se optaría por configurar un acceso mediante OpenVPN y certificados.

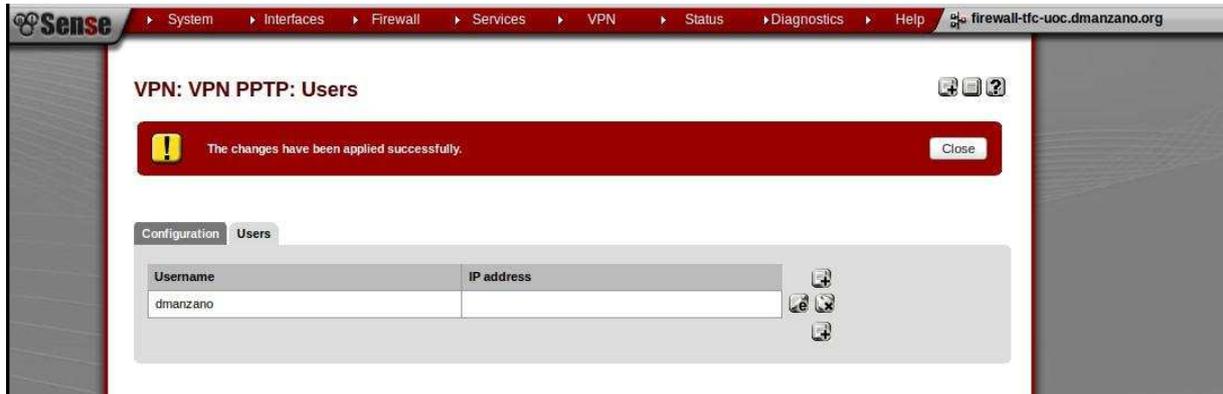
Para configurar la VPN accederemos a la interfaz web de nuestro Pfsense al apartado VPN:



Aquí como vemos configuramos una nueva red, que será la subred a la que pertenecerán los equipos que se conecten a nuestra VPN y esto nos permitirá describir reglas específicas para estos equipo, de forma que podremos decidir que permisos y hasta donde pueden acceder estos equipos. Para ello nos aparecerá una nueva pestaña en la sección de firewall que nos va a dejar incluir las reglas de control, en nuestro caso, dejamos que los equipos conectados tengan acceso tanto al DMZ como al BACKEND de nuestra empresa.



Y además generaremos el usuario con el que podremos conectar a dicha VPN. Hay que tener en cuenta que estos usuarios no podrán estar integrados en el directorio LDAP que hemos configurado, puesto que éste servicio únicamente admite autenticación mediante RADIUS.



Y con esto ya podremos acceder desde cualquier punto de internet a nuestro equipo, simplemente debemos publicar el puerto de salida de la VPN al exterior en nuestro firewall, local en nuestro caso lo hacemos gracias a la siguiente regla:

	IPv4 TCP	*	*	WAN address	1723 (PPTP)	*	none	Permitido acceso VPN	
--	-------------	---	---	-------------	----------------	---	------	-------------------------	--

Aunque PfSense lo hace automáticamente, aprovechamos para comentar que cada vez que añadimos una regla al firewall, el genera una regla de NAT en sus tablas que son las que van a realizar el enrutamiento adecuado del puerto desde el exterior hasta el destino, y esto ocurre para todas las reglas que definamos que involucren redirección de puertos o direcciones entre redes.

3.6 Bases de datos

3.6.1 MySQL

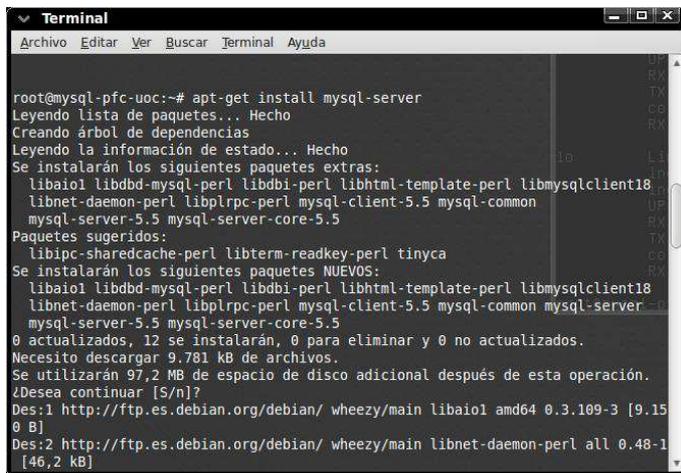


MySQL es la base de datos de código abierto más popular del mundo. Ayuda a la empresa a ofrecer unas aplicaciones de bases de datos escalables de alto rendimiento, de una forma rentable.

En abril de 2009, MySQL ofrece MySQL 5.1 en dos variantes diferentes: el código abierto MySQL Community Server y Enterprise Server comercial. MySQL 5.5 se ofrece bajo las mismas licencias.

MySQL, como la mayoría de las bases de datos relacionales transaccionales, está fuertemente limitado por el rendimiento del disco duro. Por ello en la mayoría de las ocasiones lo conveniente es realizar la instalación del mismo en servidores con almacenamiento SSD que ofrezcan un I/O alto, lo cual ayudará a reducir las latencias en todas las consultas que realicemos sobre la base de datos.

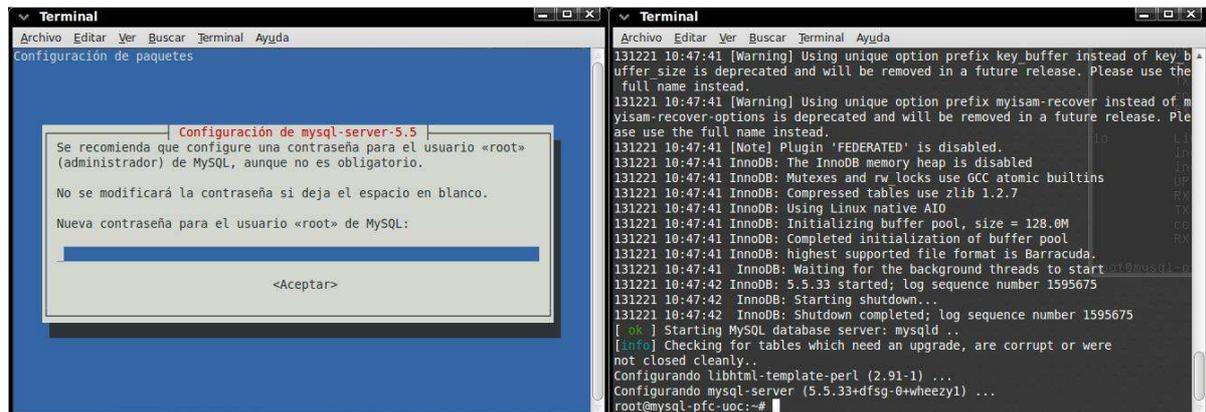
Para nuestra empresa de desarrollo, hemos creado una máquina virtual que será la encargada de alojar la base de datos. Ésta mantendrá en un sitio único todas las bases de datos de la empresa, dentro de nuestro BACKEND y accesibles para el resto de servidores que puedan necesitar conectar con la misma.



```
root@mysql-pfc-uoc:~# apt-get install mysql-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 libiaol1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
 libnet-daemon-perl liblprc-perl mysql-client-5.5 mysql-common
 mysql-server-5.5 mysql-server-core-5.5
Paquetes sugeridos:
 libipc-sharedcache-perl libterm-readkey-perl tinyca
Se instalarán los siguientes paquetes NUEVOS:
 libiaol1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
 libnet-daemon-perl liblprc-perl mysql-client-5.5 mysql-common mysql-server-5.5
mysql-server-5.5 mysql-server-core-5.5
0 actualizados, 12 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 9.781 kB de archivos.
Se utilizarán 97,2 MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
Des:1 http://ftp.es.debian.org/debian/ wheezy/main libiaol1 amd64 0.3.109-3 [9.15
0 B]
Des:2 http://ftp.es.debian.org/debian/ wheezy/main libnet-daemon-perl all 0.48-1
[46,2 kB]
```

Para desplegar, Debian nos lo facilita mediante sus repositorios de la siguiente manera:

Tras la descarga e instalación el sistema inicia el motor de bases de datos automáticamente tras preguntarnos por el usuario administrador:



```
Configuración de paquetes

Configuración de mysql-server-5.5
Se recomienda que configure una contraseña para el usuario «root»
(administrador) de MySQL, aunque no es obligatorio.

No se modificará la contraseña si deja el espacio en blanco.

Nueva contraseña para el usuario «root» de MySQL:
[ ]

<Aceptar>

131221 10:47:41 [Warning] Using unique option prefix key_buffer instead of key_b
uffer_size is deprecated and will be removed in a future release. Please use the
full name instead.
131221 10:47:41 [Warning] Using unique option prefix myisam-recover instead of m
yisam-recover-options is deprecated and will be removed in a future release. Ple
ase use the full name instead.
131221 10:47:41 [Note] Plugin 'FEDERATED' is disabled.
131221 10:47:41 InnoDB: The InnoDB memory heap is disabled
131221 10:47:41 InnoDB: Mutexes and rw locks use GCC atomic builtins
131221 10:47:41 InnoDB: Compressed tables use zlib 1.2.7
131221 10:47:41 InnoDB: Using Linux native AIO
131221 10:47:41 InnoDB: Initializing buffer pool, size = 128.0M
131221 10:47:41 InnoDB: Completed initialization of buffer pool
131221 10:47:41 InnoDB: highest supported file format is Barracuda.
131221 10:47:41 InnoDB: Waiting for the background threads to start
131221 10:47:42 InnoDB: 5.5.33 started; log sequence number 1595675
131221 10:47:42 InnoDB: Starting shutdown...
131221 10:47:42 InnoDB: Shutdown completed; log sequence number 1595675
[ OK ] Starting MySQL database server: mysqld ...
[Info] Checking for tables which need an upgrade, are corrupt or were
not closed cleanly...
Configurando libhtml-template-perl (2.91-1) ...
Configurando mysql-server (5.5.33+dfsg-0+wheezy1) ...
root@mysql-pfc-uoc:~#
```

Y ya tenemos nuestra base datos instalada e iniciada con la configuración básica, que podremos modificar dentro del `/etc/mysql/my.cnf` . En cualquier caso para nuestros propósitos, la configuración base es suficiente y ya podemos conectar y probar a crear una base de datos de ejemplo:

```
root@mysql-pfc-uoc:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 5.5.33-0+wheezy1 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database ejemplodbpfc
-> ;
Query OK, 1 row affected (0.00 sec)

mysql> |
```

Por último indicaremos que para poder conectar desde máquinas remotas al servidor, será necesario modificar el archivo de configuración por defecto, para bindear el puerto de escucha a la dirección 0.0.0.0 en lugar del al 127.0.0.1 como viene por defecto, y aparte deberemos crear el usuario especificando la IP remota para cada conexión que queramos realizar.

Ahora nuestro servidor ya está preparado para alojar las bases de datos de todos nuestros desarrollos y aplicaciones.

3.7 Publicación

3.7.1 Servidor web: Apache



El Proyecto Apache HTTP Server es un esfuerzo para desarrollar y mantener un servidor HTTP de código abierto para sistemas operativos modernos, incluyendo UNIX y Windows NT. El objetivo de este proyecto es proporcionar un servidor seguro, eficiente y extensible que proporcione servicios HTTP en sincronización con los estándares HTTP actuales.

El proyecto está gestionado conjuntamente por un grupo de voluntarios ubicados en todo el mundo, usando Internet y la Web para comunicarse, planear y desarrollar el servidor y su documentación relacionada. Además, cientos de usuarios han contribuido con ideas, código y la documentación para el proyecto.

Apache Software existe para proporcionar implementaciones robustas y de calidad comercial, de referencia de muchos tipos de software. Debe seguir siendo una plataforma sobre la cual los individuos y las instituciones pueden construir sistemas confiables, tanto con fines experimentales como para fines de misión crítica.

La Fundación de Software Apache es una entidad orgánica; los que se benefician de este programa mediante el uso de ella, a menudo contribuyen de nuevo a él, proporcionando mejoras de características, correcciones de errores y soporte para otros en las listas públicas y grupos de noticias. Este tipo de comunidades sólo pueden ocurrir gracias al Software Libre.

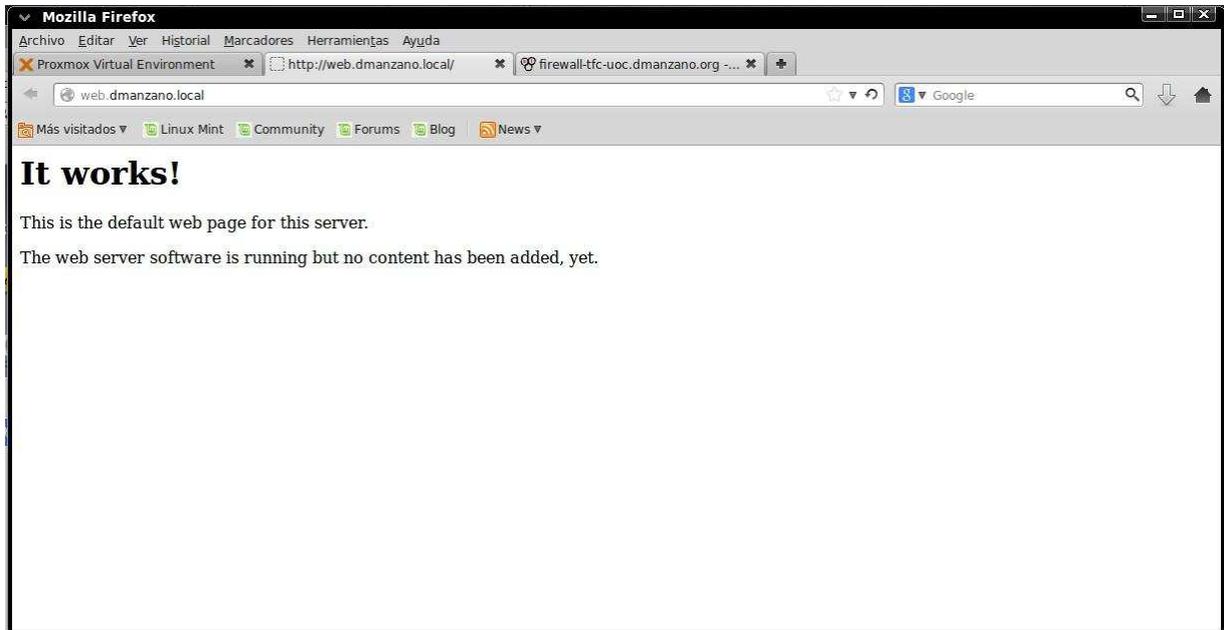
Por todo esto, Apache Web Server es el servidor escogido para ser implementado en nuestra empresa de desarrollo. Como siempre lo tenemos disponible en los repositorios oficiales de Debian. Y su instalación es sumamente sencilla:

Instalamos con el siguiente comando.

```
root@web-pfc-uoc:~# apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
Paquetes sugeridos:
  apache2-doc apache2-suexec apache2-suexec-custom openssl-blacklist
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
0 actualizados, 10 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 1.487 kB de archivos.
Se utilizarán 5.671 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? [ ]
```

```
Configurando apache2-mpm-worker (2.2.22-13) ...
[ ok ] Starting web server: apache2.
Configurando apache2 (2.2.22-13) ...
Configurando ssl-cert (1.0.32) ...
root@web-pfc-uoc:~#
```

Y al terminar, el mismo servidor arranca el proceso, de modo que ya podemos acceder a la web por defecto publicada:



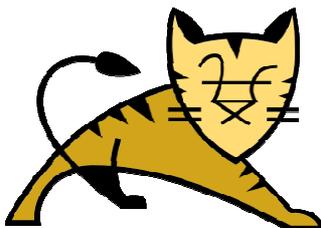
A partir de aquí, nuestro trabajo consiste en trabajar en la configuración de este servidor para adaptarlo a nuestro entorno, haciendo de este servidor el punto de entrada único para toda la publicación que podamos hacer desde la empresa.

Para esto utilizaremos los distintos módulos que apache nos ofrece. Para consultar los módulos disponibles y habilitarlos podemos hacerlo con:

```
root@web-pfc-uoc:~# a2enmod
Your choices are: actions alias asis auth basic auth digest authn_alias authn_an
on authn_dbd authn_dbm authn_default authn_file authnz_ldap authz_dbm authz_defa
ult authz_groupfile authz_host authz_owner authz_user autoindex cache cern meta
cgi cgid charset lite dav dav_fs dav_lock dbd deflate dir disk_cache dump_io env
expires ext filter file_cache filter_headers ident imagemap include info ldap l
og forensic mem_cache mime mime_magic negotiation proxy_ajp proxy_balancer
proxy_connect proxy_ftp proxy_http proxy_scgi reqtimeout rewrite setenvif spelli
ng ssl status substitute suexec unique_id userdir usertrack vhost_alias
Which module(s) do you want to enable (wildcards ok)?
```

Y gracias a dos de ellos, proxy_http y los virtualhosts, podemos redireccionar desde este único punto hacia cualquier servidor de nuestra intranet y publicar el contenido del mismo sin necesidad de exponer dichos servidores directamente al exterior.

3.7.2 Servidor de aplicaciones: Tomcat



Tomcat es una implementación de software de código abierto de las tecnologías Java Servlet y JavaServer Pages. Las especificaciones Java Servlet y JavaServer Pages están desarrolladas bajo el Java Community Process. Se desarrolla en un entorno abierto y participativo y publicado bajo la licencia Apache versión 2.

Puede usarse como un producto independiente con su propio servidor Web interno o mutuamente con otros servidores Web, como Apache, Netscape Enterprise Server, Microsoft Internet Information Server (IIS) y Microsoft Personal Web Server.

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
root@tomcat-pfc-uoc:/opt# apt-get install tomcat
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
E: No se ha podido localizar el paquete tomcat
root@tomcat-pfc-uoc:/opt# apt-cache search tomcat
guacamole-tomcat - Tomcat-based Guacamole install with VNC support
libapache-mod-jk-doc - Documentation of libapache2-mod-jk package
libapache2-mod-jk - Apache 2 connector for the Tomcat Java servlet engine
libspring-instrument-java - modular Java/J2EE application framework - Instrumentation
libspring-security-2.0-java-doc - documentation for Spring Security 2.0
libspring-security-acl-2.0-java - modular Java/J2EE application security framework - ACL
libspring-security-core-2.0-java - modular Java/J2EE application security framework - Core
libspring-security-ntlm-2.0-java - modular Java/J2EE application security framework - NTLM
libspring-security-portlet-2.0-java - modular Java/J2EE application security framework - Portlet
libspring-security-taglibs-2.0-java - modular Java/J2EE application security framework - Taglibs
liblogback-java - flexible logging library for Java
liblogback-java-doc - flexible logging library for Java - documentation
libsolr-java - Enterprise search server based on Lucene - Java libraries
solr-common - Enterprise search server based on Lucene3 - common files
solr-jetty - Enterprise search server based on Lucene3 - Jetty integration
solr-tomcat - Enterprise search server based on Lucene3 - Tomcat integration
libtomcat-maven-plugin-java - Tomcat Maven plugin
libtomcat6 - Tomcat native library using the apache portable runtime
libservlet2.5-java - Servlet 2.5 and JSP 2.1 Java API classes
libservlet2.5-java-doc - Servlet 2.5 and JSP 2.1 Java API documentation
libtomcat6-java - Servlet and JSP engine -- core libraries
tomcat6 - Servlet and JSP engine
tomcat6-admin - Servlet and JSP engine -- admin web applications
tomcat6-common - Servlet and JSP engine -- common files
tomcat6-docs - Servlet and JSP engine -- documentation
```

Tomcat server proporciona capacidades robustas así como extensas opciones de personalización, y es por esto por lo que se ha convertido en una plataforma de aplicaciones tan popular.

El beneficio más importante del servidor Tomcat es la flexibilidad. Por ejemplo, si desea ejecutar Apache en un servidor físico, pero el servicio Tomcat, el actual Tomcat JSP y servlets en otra máquina, se puede, gracias al módulo AJP que permite redireccionar tráfico entre ambos servicios.

Para la configuración de Tomcat procedemos como de costumbre a descargarnos el paquete de los repositorios oficiales de Debian como vemos en esta imagen.

```
root@tomcat-pfc-uoc:/opt# apt-get install tomcat7-admin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
 tomcat7-admin
0 actualizados, 1 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 50,1 kB de archivos.
Se utilizarán 190 kB de espacio de disco adicional después de esta operación.
Des:1 http://ftp.es.debian.org/debian/ wheezy/main tomcat7-admin all 7.0.28-4 [50,1 kB]
Descargados 50,1 kB en 5seg. (9.446 B/s)
Seleccionando el paquete tomcat7-admin previamente no seleccionado.
(Leyendo la base de datos ... 30843 ficheros o directorios instalados actualmente ...)
Desempaquetando tomcat7-admin (de .../tomcat7-admin_7.0.28-4_all.deb) ...
Configurando tomcat7-admin (7.0.28-4) ...
root@tomcat-pfc-uoc:/opt#
```

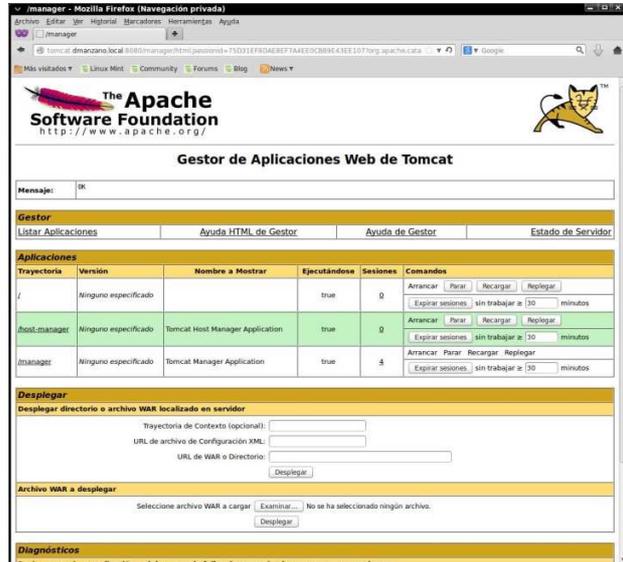
Además para poder gestionar y desplegar aplicaciones programadas en Java en el mismo, vamos a descargarnos el administrador de tomcat con el siguiente comando:

Y una vez descargado, vamos a iniciar nuestro servidor de tomcat:

```
root@tomcat-pfc-uoc:/opt# /etc/init.d/tomcat7 start
[ ok ] Starting Tomcat servlet engine: tomcat7 (already running).
root@tomcat-pfc-uoc:/opt#
```

pudiendo acceder al manager a través de la interfaz web:

Esta interfaz nos permitirá desplegar nuestras aplicaciones y desarrollos sin necesidad de tan siquiera acceder a la consola para ejecutar comandos, así como controlar el estado general de nuestro servidor de aplicaciones y modificar algunos de sus parámetros.



3.8 Gestión de proyectos

Uno de los pilares fundamentales de una empresa de desarrollo de software como es la nuestra será la consecución de proyectos software y todo proyecto necesita llevar una gestión y ser controlado, así como herramientas de apoyo al desarrollo que nos “hagan la vida más fácil”.

Por ello vamos a incluir la instalación de una serie de herramientas que nos ayudarán a llevar a conseguir estos objetivos.

3.8.1 Redmine

Redmine es una aplicación web flexible para la gestión de proyectos. Escrito utilizando el framework Ruby on Rails, es multiplataforma y cross-database.

Redmine está construido y mantenido por una comunidad de voluntarios, y será la pieza fundamental en torno a la que se orquestará todo el trabajo de equipo y desarrollo de nuestra empresa.

La instalación de Redmine, incluye la instalación de Apache y de Mysql, en este caso no utilizaremos el mysql propio de la empresa, puesto que por mera organización escogemos que Redmine utilice su propio servidor de mysql independiente y que reciba las copias de seguridad sincronizadas con el mismo.

Implementación de GNU/Linux para una empresa de desarrollo de software

David Manzano Fiestas

Para dicha instalación instalaremos los siguientes paquetes:

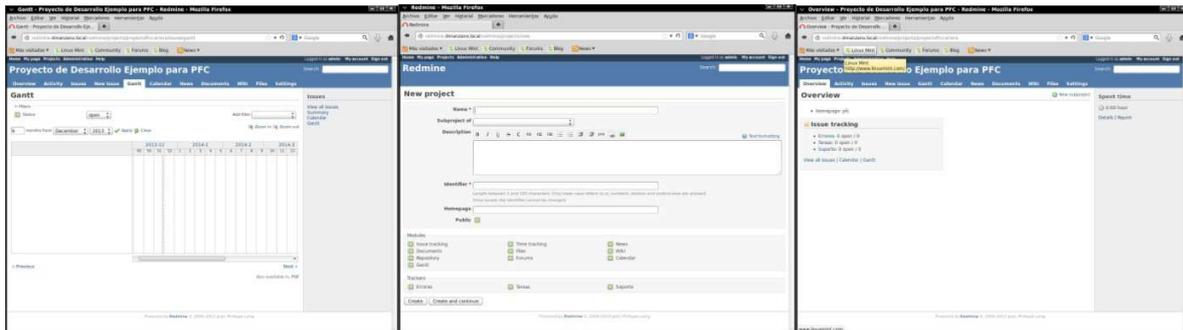
```
root@redmine-pfc-uoc:~# apt-get install redmine
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
binutils build-essential bundler cpp cpp-4.7 dbconfig-common dpkg-dev
fakeroot g++ g++-4.7 gcc gcc-4.7 javascript-common libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libc-dev-bin libc-dev
libdpkg-perl libfile-fcntllock-perl libgomp1 libltdl libltdl-dev
libjs-scriptaculous libmpc2 libmpfr4 libquadmath0 libstdc++6-4.7-dev
libtimedate-perl linux-libc-dev make manpages-dev rake ruby-actionmailer-2.3
ruby-actionpack-2.3 ruby-activerecord-2.3 ruby-activesupport-2.3
ruby-builder ruby-buildpack-ruby ruby-coderay ruby-dev
ruby-fastexecv ruby-i18n ruby-memcache-client ruby-net-ldap ruby-rack
ruby-rails-2.3 ruby-rchardet ruby-text-format ruby-tmail ruby-tzinfo
ruby1.9.1-dev rubygems-integration sudo unzip wwwconfig-common zip
Paquetes sugeridos:
binutils-doc cpp-doc gcc-4.7-locales debian-keyring g++-multilib
g++-4.7-multilib gcc-4.7-doc libstdc++6-4.7-dbg gcc-multilib autoconf
automake1.9 libtool flex bison gdb gcc-doc gcc-4.7-multilib
libmudflap0-4.7-dev libgcc1-dbg libgomp1-dbg libitm1-dbg libquadmath0-dbg
libmudflap0-dbg libctool-ppl0 libltdl-c2 libltdl7 binutils-gold apache2 httpd
glibc-doc libstdc++6-4.7-dev make-doc ruby-repack libruby-openssl bzip2 cvs darcs
git mercurial subversion rubygems postgresql-client
Se instalarán los siguientes paquetes NUEVOS:
binutils build-essential bundler cpp cpp-4.7 dbconfig-common dpkg-dev
fakeroot g++ g++-4.7 gcc gcc-4.7 javascript-common libalgorithm-diff-perl
libalgorithm-diff-xs-perl libalgorithm-merge-perl libc-dev-bin libc-dev
libdpkg-perl libfile-fcntllock-perl libgomp1 libltdl libltdl-dev
libjs-scriptaculous libmpc2 libmpfr4 libquadmath0 libstdc++6-4.7-dev
libtimedate-perl linux-libc-dev make manpages-dev rake redmine
ruby-actionmailer-2.3 ruby-actionpack-2.3 ruby-activerecord-2.3
ruby-activesupport-2.3 ruby-builder ruby-buildpack-ruby-builder
ruby-coderay ruby-dev ruby-fastexecv ruby-i18n ruby-memcache-client
ruby-net-ldap ruby-rack ruby-rails-2.3 ruby-rchardet ruby-text-format
ruby-tmail ruby-tzinfo ruby1.9.1-dev rubygems-integration sudo unzip
wwwconfig-common zip
0 actualizados, 60 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 40,2 MB de archivos.
Se utilizarán 128 MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

```
root@redmine-pfc-uoc:~# apt-get install redmine-mysql
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
libdbd-mysql-perl libdbi-perl libffi5 libmysql-ruby libmysqlclient18
libnet-daemon-perl liblprc-perl libreadline5 libruby1.8 libruby1.9.1
libyaml-0-2 mysql-client mysql-client-5.5 mysql-common redmine-mysql
ruby1.9.1
Paquetes sugeridos:
libterm-readkey-perl mysql-server ri ruby-dev ruby1.9.1-examples ri1.9.1
graphviz ruby1.9.1-dev ruby-switch
Se instalarán los siguientes paquetes NUEVOS:
libdbd-mysql-perl libdbi-perl libffi5 libmysql-ruby libmysqlclient18
libnet-daemon-perl liblprc-perl libreadline5 libruby1.8 libruby1.9.1
libyaml-0-2 mysql-client mysql-client-5.5 mysql-common redmine-mysql
ruby-mysql ruby1.9.1
0 actualizados, 18 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 10,9 MB de archivos.
Se utilizarán 64,4 MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

```
root@redmine-pfc-uoc:~# apt-get install libapache2-mod-passenger
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
libapache2-mod-passenger
0 actualizados, 1 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 246 kB de archivos.
Se utilizarán 799 kB de espacio de disco adicional después de esta operación.
Des:1 http://ftp.es.debian.org/debian/ wheezy/main libapache2-mod-passenger amd64
3.0.13debian-1+deb7u1 [246 kB]
Descargados 246 kB en 5seg. (44,7 kB/s)
Seleccionando el paquete libapache2-mod-passenger previamente no seleccionado.
(Leyendo la base de datos ... 39898 ficheros o directorios instalados actualmente)
Desempaquetando libapache2-mod-passenger (de .../libapache2-mod-passenger_3.0.13deb
ebian-1+deb7u1 amd64.deb) ...
Configurando libapache2-mod-passenger (3.0.13debian-1+deb7u1) ...
[ ok ] Reloading web server config: apache2.
root@redmine-pfc-uoc:~#
```

Una vez instalado, vamos a Acceder via web y ver como ya tenemos nuestro gestor de proyectos operativo:

Aquí tenemos algunas pantallas que muestran cómo creamos un proyecto de ejemplo y distintas pantallas de control de la aplicación:



Redmine pone a nuestra disposición entre otros elementos, un sistema de Bug Tracking o una Wiki, por lo que es más que probable que queramos mantener este sistema publicado al exterior para que el cliente pueda incluir comentarios o acceder a información de uso de la aplicación que habremos desarrollado para él. Ésto como

siempre se hará a través del Apache que tenemos situado en nuestro DMZ evitando la publicación directa de este entorno al exterior.

3.8.4 Netbeans y Eclipse

Para que los desarrolladores de nuestra empresa puedan llevar a cabo su labor de la forma más eficiente posible, debemos trabajar con sistemas que nos permitan realizar una programación multilinguaje. Para ellos el software libre pone a nuestro disposición dos grandes herramientas como son Netbeans y Eclipse, ambas se puede obtener directamente desde los repositorios de Debian, y tanto una como la otra nos ofrecen toda la potencia y versatilidad necesaria para acometer los desarrollos que sean necesarios.

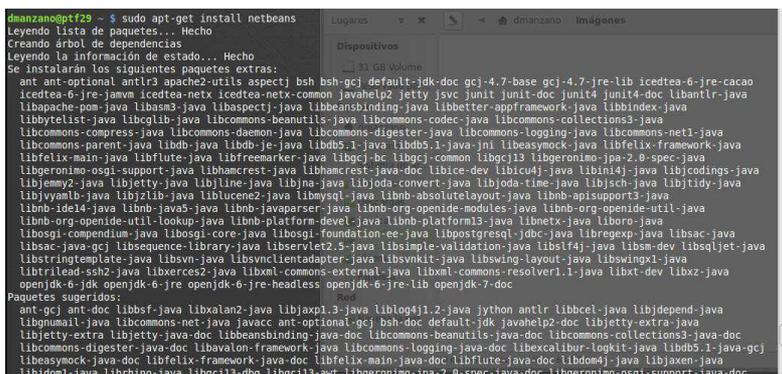
- **NetBeans** es un entorno de desarrollo integrado (IDE) para el desarrollo principalmente con Java, pero también con otros idiomas, en particular, PHP, C / C + +, y HTML5. También es marco plataforma de aplicaciones para aplicaciones de escritorio Java y otros.

NetBeans administra las siguientes características de la plataforma y componentes:

- Ajustes del usuario
- Ventanas (colocación, apariencia, etc)
- NetBeans Visual Library
- Almacenamiento
- Herramientas de desarrollo integradas
- Marco asistente

El IDE NetBeans está escrito en Java y puede ejecutarse en Windows, OS X, Linux, Solaris y otras plataformas que soporten una JVM compatible. La plataforma NetBeans permite que las aplicaciones se desarrollen a partir de un conjunto de componentes modulares de software llamados módulos.

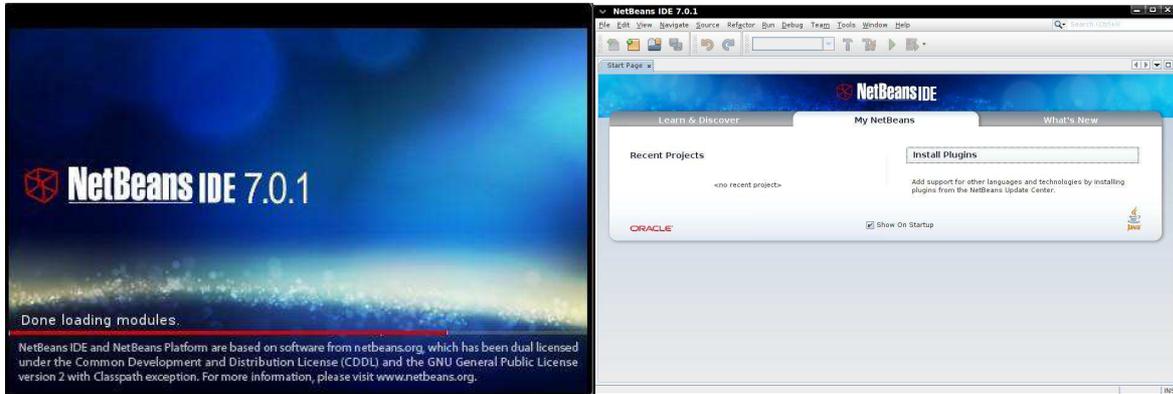
El Equipo NetBeans da soporte de forma activa al producto, y busca sugerencias de la comunidad en general. Cada versión está precedida por un tiempo para las pruebas y Sugerencias por parte de la Comunidad.



```
dmanzano@ptf29 ~ $ sudo apt-get install netbeans
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 ant ant-optional antlr3 apache2-utils aspectj bsh bsh-gcj default-jdk-doc gcj-4.7-base gcj-4.7-jre-lib icedtea-6-jre-cacao
icedtea-6-jre-javmvm icedtea-netx icedtea-netx-common javahelp2 jetty jsvc junit junit-doc junit4 junit4-doc libantlr-java
libapache-pom-java libasm3-java libaspectj-java libbeansbinding-java libbetter-aptframework-java libbindex-java
libbytestlist-java libcglib-java libcommons-beanutils-java libcommons-codec-java libcommons-collections3-java
libcommons-compress-java libcommons-daemon-java libcommons-digester-java libcommons-logging-java libcommons-net1-java
libcommons-parent-java libdb-java libdb-je-java libdb5.1-java libdb5.1-java-jni libeasymock-java libfelix-framework-java
libfelix-main-java libfute-java libfreemarker-java libgcj-bc libgcj-common libgcj13 libgeronimo-jpa-2.0-spec-java
libgeronimo-osgi-support-java libhamcrest-java libhamcrest-java-doc libice-dev libicu4j-java libin4j-java libjcodings-java
libjcommon2-java libjetty-java libjline-java libjna-java libjoda-convert-java libjoda-time-java libjsch-java libjtidy-java
libjvarkit-java libjzlib-java liblucene2-java libmysql-java libnb-absolutelayout-java libnb-apisupport3-java
libnb-ide14-java libnb-javaws-java libnb-javaparser-java libnb-org-openide-modules-java libnb-org-openide-util-java
libnb-org-openide-util-lookup-java libnb-platform-devel-java libnb-platform3-java libnetx-java liboro-java
libosgi-compendium-java libosgi-core-java libosgi-foundation-ee-java libpostgresql-jdbc-java libregexp-java libsac-java
libsac-java-gcj libsequence-library-java libservlet2.5-java libsimple-validation-java libslf4j-java libsm-dev libsqljet-java
libstringtemplate-java libsvn-java libsvnclientadapter-java libsvnkit-java libswing-layout-java libswingx1-java
libtrilead-ssh2-java libxerces2-java libxml-commons-external-java libxml-commons-resolver1.1-java libxt-dev libxz-java
openjdk-6-jdk openjdk-6-jre openjdk-6-jre-headless openjdk-6-jre-lib openjdk-7-doc
Paquetes sugeridos:
 ant-gcj ant-doc libbsf-java libxalan2-java libjaxp1.3-java liblog4j1.2-java jython antlr libbcel-java libjdepend-java
libjmail-java libcommons-net-java javacc ant-optional-gcj bsh-doc default-jdk javahelp2-doc libjetty-extra-java
libjetty-extra libjetty-java-doc libbeansbinding-java-doc libcommons-beanutils-java-doc libcommons-collections3-java-doc
libcommons-digester-java-doc libavalon-framework-java libcommons-logging-java-doc libxerceslib-logging-java libdb5.1-java-gcj
libeasymock-java-doc libfelix-framework-java-doc libfelix-main-java-doc libfute-java-doc libdom4j-java libjaxen-java
libjdom1-java librhino-java libgcj13-dbg libgcj13-awt libgeronimo-jpa-2.0-spec-java-doc libgeronimo-osgi-support-java-doc
```

Su instalación en nuestro equipo de desarrollo de pruebas es como siempre extremadamente simple:

Y ya estamos listos para trabajar con ella, desde los menús del escritorio de Debian:



- **Eclipse** es un entorno de desarrollo integrado, de Código abierto y Multiplataforma. Es una plataforma de programación, desarrollo y compilación de elementos tales como sitios web, programas en C++ o aplicaciones Java, muy completa y potente.

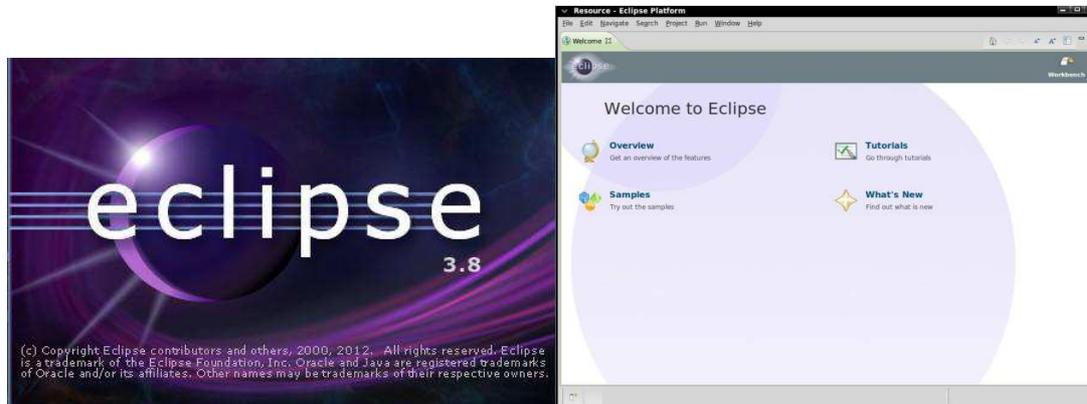
Eclipse se pueden ampliar y mejorar mediante el uso de plug-ins, a través de los cuales se puede añadir un sistema de control de versiones a través de subversion y a la vez lograr una integración mediante Hibernate. El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos para adquirir las funcionalidades que necesite el usuario (no están todas incluidas, las necesite el usuario o no).

Eclipse provee al programador con Frameworks muy completos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de Software, Aplicaciones web, etc.

Para realizar la instalación del mismo:

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
manzanogtf2@ ~$ sudo apt-get install eclipse
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  binfmt-support default-jdk eclipse-jdt eclipse-pde eclipse-platform eclipse-platform-data eclipse-rcp fastjar jarwrapper
  libcommons-cli-java libcommons-dbcp-java libcommons-httpclient-java libcommons-lang-java libcommons-pool-java libcecj-java
  libeclipse-osgi-java libfelix-bundlerepository-java libfelix-gogo-command-java libfelix-gogo-runtime-java
  libfelix-gogo-shell-java libfelix-osgi-obr-java libfelix-shell-java libfelix-utils-java libgeronimo-jta-1.1-spec-java
  libicuj4-4.4-java libjetty8-java libxml2-java libswt-cairo-gtk-3-jni libswt-gtk-gtk-3-jni libswt-gnome-gtk-3-jni
  libswt-gtk-3-java libswt-gtk-3-jni libswt-webkit-gtk-3-jni libtomcat7-java openjdk-7-jdk sat4j
Paquetes sugeridos:
  libcommons-dbcp-java-doc libcommons-httpclient-java-doc ecj libecj-java-gcj libfelix-bundlerepository-java-doc
  libfelix-gogo-command-java-doc libfelix-gogo-runtime-java-doc libfelix-gogo-shell-java-doc libfelix-osgi-obr-java-doc
  libfelix-shell-java-doc libfelix-utils-java-doc libgeronimo-jta-java-doc jetty8 libjetty8-java-doc libswt-gtk-3-java-gcj
  tomcat7 openjdk-7-demo openjdk-7-source visualvm
Se instalarán los siguientes paquetes NUEVOS:
  binfmt-support default-jdk eclipse eclipse-jdt eclipse-pde eclipse-platform eclipse-platform-data eclipse-rcp fastjar
  jarwrapper libcommons-cli-java libcommons-dbcp-java libcommons-httpclient-java libcommons-lang-java libcommons-pool-java
  libcecj-java libeclipse-osgi-java libfelix-bundlerepository-java libfelix-gogo-command-java libfelix-gogo-runtime-java
  libfelix-gogo-shell-java libfelix-osgi-obr-java libfelix-shell-java libfelix-utils-java libgeronimo-jta-1.1-spec-java
  libicuj4-4.4-java libjetty8-java libxml2-java libswt-cairo-gtk-3-jni libswt-gtk-gtk-3-jni libswt-gnome-gtk-3-jni
  libswt-gtk-3-java libswt-gtk-3-jni libswt-webkit-gtk-3-jni libtomcat7-java openjdk-7-jdk sat4j
0 actualizados, 37 se instalarán, 0 para eliminar y 50 no actualizados.
Necesito descargar 166 MB de archivos.
Se utilizarán 192 MB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? S
Des:1 http://archive.ubuntu.com/ubuntu/ raring/main libcecj-java all 3.5.1-6 [1,231 kB]
Des:2 http://archive.ubuntu.com/ubuntu/ raring/main libcommons-pool-java all 1.5.6-1build1 [94,3 kB]
Des:3 http://archive.ubuntu.com/ubuntu/ raring/main libcommons-dbcp-java all 1.4-3ubuntu1 [149 kB]
Des:4 http://archive.ubuntu.com/ubuntu/ raring-updates/main libtomcat7-java all 7.0.25-1-exp2ubuntu1.1 [3,619 kB]
Des:5 http://archive.ubuntu.com/ubuntu/ raring/universe eclipse-platform-data all 3.8.1-1ubuntu1 [34,2 MB]
Des:6 http://archive.ubuntu.com/ubuntu/ raring/universe libeclipse-osgi-java all 3.8.1-1ubuntu1 [2,697 kB]
```

Y queda listo para trabajar con el:



Esas son las dos herramientas más conocidas para el desarrollo de aplicaciones, pero GNU/Linux pone a nuestra disposición un listado imposible de cuantificar de herramientas, tanto desde su repositorio como a través de repositorios de terceros. Y todos quedan disponibles para su instalación rápida y eficiente gracias a la interfaz de línea de comandos de Debian.

3.8.5 Subversion

Subversion es un Sistema de Control de Versiones de código abierto. Subversion ha gozado y sigue gozando de la adopción generalizada tanto en el campo de código abierto como en el mundo empresarial. Además forma parte de una rica comunidad de desarrolladores y usuarios.

La estructura habitual de un repositorio de Subversion es:

- Trunk: desarrollo principal.
- Tags: ubicación de las versiones congeladas.
- Branches: ubicación con versiones de desarrollo paralelas al trunk.

La utilización de Subversion en los desarrollos de nuestra empresa, será necesaria y crítica puesto que hoy en día es prácticamente imposible imaginar una empresa de desarrollo en la que los desarrollos se realizan de forma faseada y colaborativa, si que se utilice un sistema de este tipo. Además tanto Eclipse como Netbeans que son las dos herramientas de desarrollo elegidas proporcionan integración directa con subversion.

La instalación de subversion se realiza a través de los repositorios y lo hacemos sirviendo el mismo gracias a Apache y uno de los módulos del mismo:

Implementación de GNU/Linux para una empresa de desarrollo de software

David Manzano Fiestas

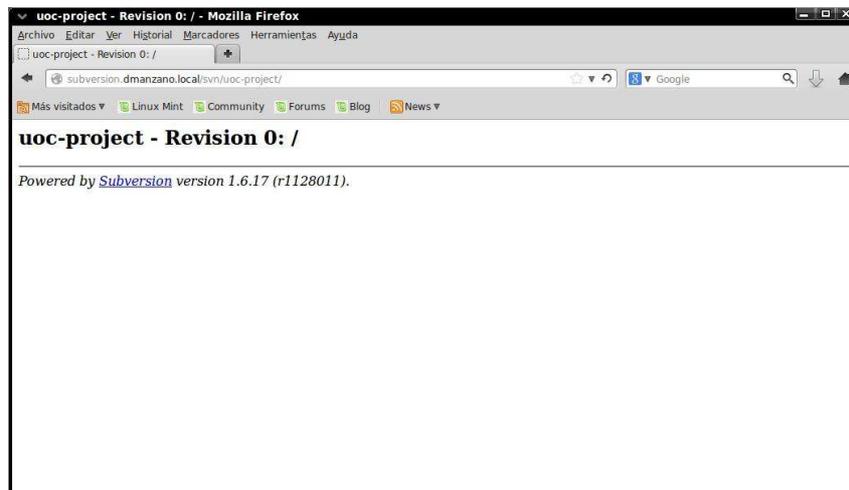
```
root@subversion-pfc-uoc:~# apt-get install subversion
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libapr1 libaprutil1 libneon27-gnutls libsvn1
Paquetes sugeridos:
  subversion-tools
Se instalarán los siguientes paquetes NUEVOS:
  libapr1 libaprutil1 libneon27-gnutls libsvn1 subversion
0 actualizados, 5 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 2.591 kB de archivos.
Se utilizarán 7.076 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [5/n]?
Des:1 http://ftp.es.debian.org/debian/ wheezy/main libapr1 amd64 1.4.6-3+deb7u1 [106 kB]
Des:2 http://ftp.es.debian.org/debian/ wheezy/main libaprutil1 amd64 1.4.1-3 [89,0 kB]
Des:3 http://ftp.es.debian.org/debian/ wheezy/main libneon27-gnutls amd64 0.29.6-3 [142 kB]
Des:4 http://ftp.es.debian.org/debian/ wheezy/main libsvn1 amd64 1.6.17dfsg-4+deb7u4 [934 kB]
Des:5 http://ftp.es.debian.org/debian/ wheezy/main subversion amd64 1.6.17dfsg-4+deb7u4 [1.320 kB]
Descargados 2.591 kB en 0seg. (316 kB/s)
Seleccionando el paquete libapr1 previamente no seleccionado.
(Leyendo la base de datos ... 27653 ficheros o directorios instalados actualmente ...)
Desempaquetando libapr1 (de .../libapr1.1.4.6-3+deb7u1_amd64.deb) ...
Seleccionando el paquete libaprutil1 previamente no seleccionado.
Desempaquetando libaprutil1 (de .../libaprutil1.1.4.1-3_amd64.deb) ...
Seleccionando el paquete libneon27-gnutls previamente no seleccionado.
Desempaquetando libneon27-gnutls (de .../libneon27-gnutls.0.29.6-3_amd64.deb) ...
Seleccionando el paquete libsvn1:amd64 previamente no seleccionado.
Desempaquetando libsvn1:amd64 (de .../libsvn1.1.6.17dfsg-4+deb7u4_amd64.deb) ...
Seleccionando el paquete subversion previamente no seleccionado.
Desempaquetando subversion (de .../subversion.1.6.17dfsg-4+deb7u4_amd64.deb) ...
Procesando disparadores para man-db ...
Configurando libapr1 (1.4.6-3+deb7u1) ...
Configurando libaprutil1 (1.4.1-3) ...
Configurando libneon27-gnutls (0.29.6-3) ...
Configurando libsvn1:amd64 (1.6.17dfsg-4+deb7u4) ...
Configurando subversion (1.6.17dfsg-4+deb7u4) ...
root@subversion-pfc-uoc:~#
```

```
root@subversion-pfc-uoc:~# apt-get install apache2 libapache2-svn
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common
  libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
Paquetes sugeridos:
  apache2-doc apache2-suexec apache2-suexec-custom openssl-blacklist
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common
  libapache2-svn libaprutil1-dbd-sqlite3 libaprutil1-ldap ssl-cert
0 actualizados, 9 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 1.463 kB de archivos.
Se utilizarán 5.579 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [5/n]? S
Des:1 http://ftp.es.debian.org/debian/ wheezy/main libaprutil1-dbd-sqlite3 amd64 1.4.1-3 [19,0 kB]
Des:2 http://ftp.es.debian.org/debian/ wheezy/main libaprutil1-ldap amd64 1.4.1-3 [16,6 kB]
Des:3 http://ftp.es.debian.org/debian/ wheezy/main apache2.2-bin amd64 2.2.22-13 [779 kB]
Des:4 http://ftp.es.debian.org/debian/ wheezy/main apache2-utils amd64 2.2.22-13 [162 kB]
Des:5 http://ftp.es.debian.org/debian/ wheezy/main apache2.2-common amd64 2.2.22-13 [291 kB]
Des:6 http://ftp.es.debian.org/debian/ wheezy/main apache2-mpm-worker amd64 2.2.2-2-13 [2.256 B]
Des:7 http://ftp.es.debian.org/debian/ wheezy/main apache2 amd64 2.2.22-13 [1.432 B]
Des:8 http://ftp.es.debian.org/debian/ wheezy/main libapache2-svn amd64 1.6.17dfsg-4+deb7u4 [173 kB]
Des:9 http://ftp.es.debian.org/debian/ wheezy/main ssl-cert all 1.0.32 [19,5 kB]
Descargados 1.463 kB en 7seg. (204 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete libaprutil1-dbd-sqlite3 previamente no seleccionado.
(Leyendo la base de datos ... 27766 ficheros o directorios instalados actualmente ...)
Desempaquetando libaprutil1-dbd-sqlite3 (de .../libaprutil1-dbd-sqlite3.1.4.1-3_amd64.deb) ...
Seleccionando el paquete libaprutil1-ldap previamente no seleccionado.
Desempaquetando libaprutil1-ldap (de .../libaprutil1-ldap.1.4.1-3_amd64.deb) ...
Seleccionando el paquete apache2.2-bin previamente no seleccionado.
Desempaquetando apache2.2-bin (de .../apache2.2-bin.2.2.22-13_amd64.deb) ...
```

Ahora ya podemos hacer uso de subversión, para ello en nuestro portatil de desarrollo instalamos el cliente de subversion y hacemos la prueba de “checkout” contra el proyecto de prueba generado:

```
dmanzano@ptf29 ~/proyecto-uoc $ svn co http://subversion.dmanzano.local/svn/uoc-project
Checked out revision 0.
dmanzano@ptf29 ~/proyecto-uoc $
```

Y aquí tenemos el acceso web a nuestro servidor de subversion:



3.9 Otros servicios

3.9.1 DNS con Bind9

Un servidor DNS (Domain Name System) es un sistema que nos permite usar nombres de dominio en lugar de direcciones IP. Su principal ventaja es que es mucho más fácil recordar un nombre que una dirección IP.

Bind9 es posiblemente la implementación más famosa en software libre de dicho servidor.

El servidor DNS bind admite tres modos de funcionamiento:

- Servidor DNS maestro
- Servidor DNS esclavo
- Servidor caché DNS

En nuestro caso hemos implementado el servicio de DNS con zona local únicamente, es decir, solo estamos sirviendo nombres para nuestra intranet, aunque podríamos implementar zona global y gestionar cualquier dominio externo mediante este servidor.

Si nos hemos fijado durante todas las pantallas de instalación y accesos web anteriores podemos comprobar como hemos estado accediendo a las diferentes máquinas de nuestra intranet mediante nombres en lugar de hacerlo mediante direcciones IP, esto se debe a la implementación de este servidor de DNS en el DMZ de nuestra red.

Para esta implementación, lo hemos hecho a través de los repositorios de Debian de la siguiente manera:

```
root@dns-pfc-uoc:~# apt-get install bind9 bind9utils bind9-doc dnsutils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
dnsutils ya está en su versión más reciente.
Paquetes sugeridos:
  resolvconf ufw
Se instalarán los siguientes paquetes NUEVOS:
  bind9 bind9-doc bind9utils
0 actualizados, 3 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 864 kB de archivos.
Se utilizarán 2.588 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? █
```

Y luego generamos el forward y el reverse de nuestra zona a partir de los archivos de configuración base:

```
root@dns-pfc-uoc:~# vim /etc/bind/named.conf.local
root@dns-pfc-uoc:~# cp /etc/bind/db.local /etc/bind/for.dmanzano.local
root@dns-pfc-uoc:~# vim /etc/bind/for.dmanzano.local
root@dns-pfc-uoc:~# cp /etc/bind/db.127 /etc/bind/rev.dmanzano.local
root@dns-pfc-uoc:~# vim /etc/bind/rev.dmanzano.local
root@dns-pfc-uoc:~# service bind9 restart
[ ok ] Stopping domain name service...: bind9.
[ ok ] Starting domain name service...: bind9.
root@dns-pfc-uoc:~# named
named                named-checkconf      named-checkzone      named-compilezone    named-journalprint
root@dns-pfc-uoc:~# named
named                named-checkconf      named-checkzone      named-compilezone    named-journalprint
root@dns-pfc-uoc:~# named-checkconf /etc/bind/named.conf.local
root@dns-pfc-uoc:~# named-checkzone dmanzano.local /etc/bind/for.dmanzano.local
zone dmanzano.local/IN: loaded serial 2
OK
root@dns-pfc-uoc:~# █
```

Ahora mismo ya vemos como el servidor empieza a funcionar con una configuración básica, en la que ya resuelve nuestra zona local y los nombres de internet:

```
root@dns-pfc-uoc:~# nslookup dmanzano.local
Server:          192.168.2.2
Address:         192.168.2.2#53

Name:   dmanzano.local
Address: 192.168.2.2

root@dns-pfc-uoc:~# █
```

```
root@dns-pfc-uoc:~# nslookup www.google.es
Server:          192.168.2.2
Address:         192.168.2.2#53

Non-authoritative answer:
Name:   www.google.es
Address: 173.194.34.248
Name:   www.google.es
Address: 173.194.34.247

root@dns-pfc-uoc:~# █
```

Pero lo realmente interesante será que podremos hacer que resuelva nombres internos de nuestra red, modificando el siguiente archivo de configuración vemos como el sistema empieza a funcionar:

```
Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
;
; BIND data file for forward.dmanzano.local zone
;
$TTL      604800
@         IN      SOA      master.dmanzano.local. root.dmanzano.local (
                        3          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@         IN      NS       master.dmanzano.local.
@         IN      NS       slave.dmanzano.local.
@         IN      A        192.168.2.2
@         IN      AAAA     ::1
master   IN      A        192.168.2.2
slave    IN      A        192.168.2.2
ntp      IN      A        192.168.2.4
web      IN      A        192.168.2.5
correo   IN      A        192.168.2.6
```

Ahora si hacemos un ping, nuestro DNS nos traducirá ese nombre en la IP correspondiente y esto simplificará mucho el trabajo dentro de la empresa.

```
root@dns-pfc-uoc:~# ping ntp.dmanzano.local
PING ntp.dmanzano.local (192.168.2.4) 56(84) bytes of data.
64 bytes from 192.168.2.4: icmp_req=1 ttl=64 time=0.692 ms
64 bytes from 192.168.2.4: icmp_req=2 ttl=64 time=0.300 ms
64 bytes from 192.168.2.4: icmp_req=3 ttl=64 time=0.388 ms
64 bytes from 192.168.2.4: icmp_req=4 ttl=64 time=0.374 ms
64 bytes from 192.168.2.4: icmp_req=5 ttl=64 time=0.361 ms
64 bytes from 192.168.2.4: icmp_req=6 ttl=64 time=0.394 ms
64 bytes from 192.168.2.4: icmp_req=7 ttl=64 time=0.367 ms
^C
--- ntp.dmanzano.local ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 7324ms
rtt min/avg/max/mdev = 0.300/0.410/0.692/0.121 ms
root@dns-pfc-uoc:~#
```

3.9.2 Sincronización horaria con NTP

NTP es un protocolo de Internet para sincronizar los relojes de los sistemas informáticos a través del enrutamiento de paquetes en redes con latencia variable. NTP utiliza UDP como su capa de transporte, usando el puerto 123. Está diseñado para resistir los efectos de la latencia variable.

Debian GNU/Linux dispone de su servidor NTP que se instala y configura de una forma extremadamente sencilla, y aunque parece algo que no es muy importante, cabe destacar la importancia de mantener sincronizados todos los servidores de un CPD, puesto que como lo normal es que unos servidores interactúen con otros, si no se encuentran sincronizados en fecha y hora esto puede provocar situaciones insostenibles para muchas aplicaciones como es el hecho de que se pasen datos de unos servidores a otros con fechas del futuro o del pasado.

Para evitar esta situación vamos a implementar nuestro propio servidor de sincronización:

```
root@ntp-pfc-uoc:~# aptitude install ntp
Se instalarán los siguiente paquetes NUEVOS:
libopts25{a} ntp
0 paquetes actualizados, 2 nuevos instalados, 0 para eliminar y 0 sin actualizar.
Necesito descargar 635 kB de ficheros. Después de desempaquetar se usarán 1.460 kB.
¿Quiere continuar? [Y/n/?]
```

Y ahora revisamos el archivo de configuración donde decimos que los servidores y usuarios de nuestra intranet pueden contactar con el para sincronizar:

```

/etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help
driftfile /var/lib/ntp/ntp.drift

# Enable this if you want statistics to be logged.
#statsdir /var/log/ntpstats/

statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

# You do need to talk to an NTP server or two (or three).
#server ntp.your-provider.example

# pool.ntp.org maps to about 1000 low-stratum NTP servers.
# pick a different set every time it starts up. Please consider joining the
# pool: <http://www.pool.ntp.org/join.html>
server 0.debian.pool.ntp.org iburst
server 1.debian.pool.ntp.org iburst
server 2.debian.pool.ntp.org iburst
server 3.debian.pool.ntp.org iburst

# Access control configuration; see /usr/share/doc/ntp-doc/html/accept.html for
# details. The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>
# might also be helpful.
#
# Note that "restrict" applies to both servers and clients, so a configuration
# that might be intended to block requests from certain clients could also end
# up blocking replies from your own upstream servers.

# By default, exchange time with everybody, but don't allow configuration.
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1

# Clients from this (example!) subnet have unlimited access, but only if
# cryptographically authenticated.
#restrict 192.168.123.0 mask 255.255.255.0 notrust

# If you want to provide time to your local subnet, change the next line.
# (Again, the address is an example only.)
#broadcast 192.168.123.255

# If you want to listen to time broadcasts on your local subnet, de-comment the
# next lines. Please do this only if you trust everybody on the network!
#disable auth
#broadcastclient
    
```

Reiniciamos el servicio:

```

root@ntp-pfc-uoc:~# /etc/init.d/ntp restart
[ ok ] Stopping NTP server: ntpd.
[ ok ] Starting NTP server: ntpd.
root@ntp-pfc-uoc:~# vim /etc/ntp.conf
    
```

Y comprobamos que funciona:

```

root@ntp-pfc-uoc:~# ntpq -p
      remote           refid      st t when poll reach   delay   offset  jitter
-----
-dns3.masbytes.e 173.230.149.23  3 u  42  64   1  58.824 -1.687  1.772
+142.54.181.202  204.9.54.119   2 u  41  64   1 185.799  2.454  5.804
*dnscache-madrid 140.203.204.77  2 u  41  64   1  82.736 20.169  0.630
+PTV.EXA-PTG.ES  150.214.94.5   2 u  40  64   1  61.883  4.976  8.819
    
```

```

root@ntp-pfc-uoc:~# ntpdate ntp.dmanzano.local
19 Dec 21:02:08 ntpdate[4499]: adjust time server 192.168.2.4 offset 0.000707 sec
    
```

bastará con añadir esa línea al crontab de nuestros servidores para que periódicamente sincronicen con este y mantengamos toda nuestra granja de servidores plenamente sincronizada.

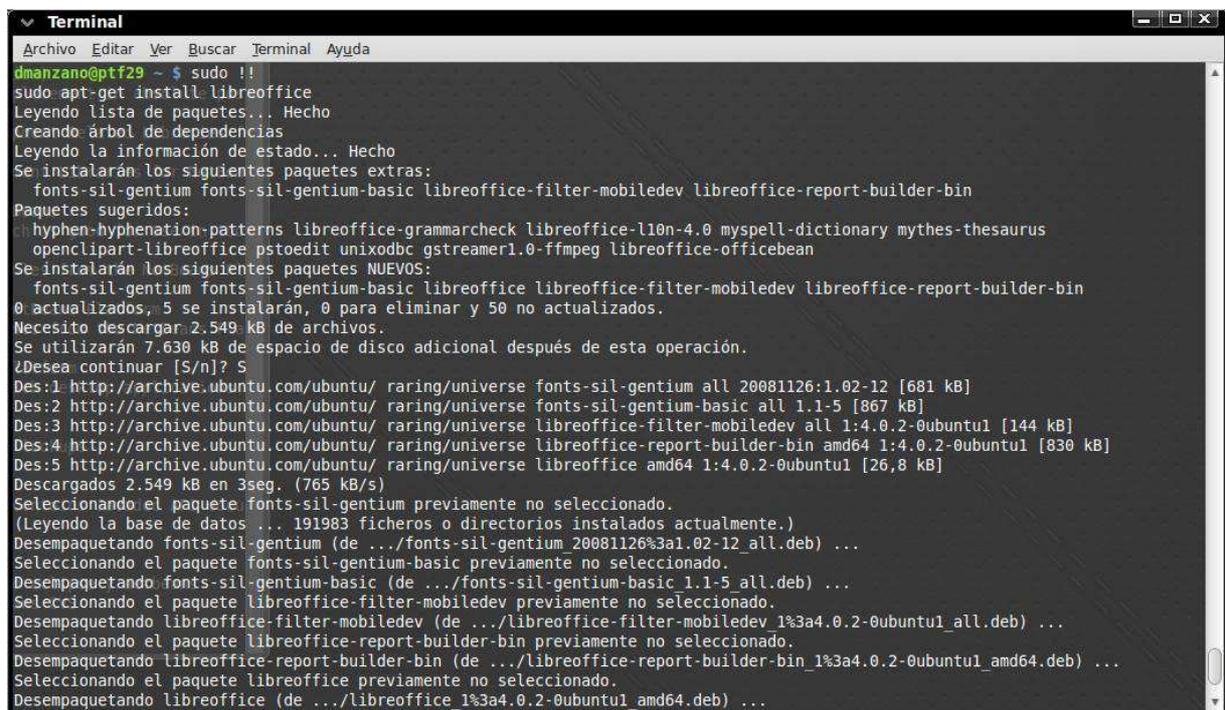
3.9.3 Libreoffice

Poco se podemos hablar ya de esta suite ofimática libre y de código abierto desarrollada por The Document Foundation. Se creó como bifurcación de OpenOffice.org en 2010. Tiene un procesador de texto (*Writer*), un editor de hojas de cálculo (*Calc*), un gestor de presentaciones (*Impress*), un gestor de bases de datos (*Base*), un editor de gráficos vectoriales (*Draw*) y un editor de fórmulas matemáticas (*Math*).

Es compatible con las principales suites ofimáticas, incluyendo Microsoft Office, aunque algunas características de diseño y atributos de formato son manejados de forma diferente o no están soportados.

Está disponible para diferentes sistemas operativos incluyendo Microsoft Windows, GNU/Linux y Mac OS y por ello es la que utilizaremos en nuestra empresa de desarrollo para toda la labor ofimática,

La instalación en nuestros sistemas es como siempre a partir de los repositorios de Debian:



```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
dmanzano@ptf29 ~ $ sudo !!
sudo apt-get install libreoffice
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
 fonts-sil-gentium fonts-sil-gentium-basic libreoffice-filter-mobiledev libreoffice-report-builder-bin
Paquetes sugeridos:
 hyphen-hyphenation-patterns libreoffice-grammarcheck libreoffice-l10n-4.0 myspell-dictionary mythes-thesaurus
 openclipart-libreoffice pstoeedit unixodbc gstreamer1.0-ffmpeg libreoffice-officebean
Se instalarán los siguientes paquetes NUEVOS:
 fonts-sil-gentium fonts-sil-gentium-basic libreoffice libreoffice-filter-mobiledev libreoffice-report-builder-bin
0 actualizados, 5 se instalarán, 0 para eliminar y 50 no actualizados.
Necesito descargar 2.549 kB de archivos.
Se utilizarán 7.630 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? S
Des:1 http://archive.ubuntu.com/ubuntu/ raring/universe fonts-sil-gentium all 20081126:1.02-12 [681 kB]
Des:2 http://archive.ubuntu.com/ubuntu/ raring/universe fonts-sil-gentium-basic all 1.1-5 [867 kB]
Des:3 http://archive.ubuntu.com/ubuntu/ raring/universe libreoffice-filter-mobiledev all 1:4.0.2-0ubuntu1 [144 kB]
Des:4 http://archive.ubuntu.com/ubuntu/ raring/universe libreoffice-report-builder-bin amd64 1:4.0.2-0ubuntu1 [830 kB]
Des:5 http://archive.ubuntu.com/ubuntu/ raring/universe libreoffice amd64 1:4.0.2-0ubuntu1 [26,8 kB]
Descargados 2.549 kB en 3seg. (765 kB/s)
Seleccionando el paquete fonts-sil-gentium previamente no seleccionado.
(Leyendo la base de datos ... 191983 ficheros o directorios instalados actualmente.)
Desempaquetando fonts-sil-gentium (de ../fonts-sil-gentium_20081126%3a1.02-12_all.deb) ...
Seleccionando el paquete fonts-sil-gentium-basic previamente no seleccionado.
Desempaquetando fonts-sil-gentium-basic (de ../fonts-sil-gentium-basic_1.1-5_all.deb) ...
Seleccionando el paquete libreoffice-filter-mobiledev previamente no seleccionado.
Desempaquetando libreoffice-filter-mobiledev (de ../libreoffice-filter-mobiledev_1%3a4.0.2-0ubuntu1_all.deb) ...
Seleccionando el paquete libreoffice-report-builder-bin previamente no seleccionado.
Desempaquetando libreoffice-report-builder-bin (de ../libreoffice-report-builder-bin_1%3a4.0.2-0ubuntu1_amd64.deb) ...
Seleccionando el paquete libreoffice previamente no seleccionado.
Desempaquetando libreoffice (de ../libreoffice_1%3a4.0.2-0ubuntu1_amd64.deb) ...
```

Y ya estamos listos para trabajar con libreoffice.

4. Descripción de la implementación de la simulación

La simulación la hemos llevado a cabo utilizando ordenadores normales de sobremesa a los cuales les ha sido sustituido el disco duro, para no machacar el original y poder realizar la instalación de los distintos sistemas operativos necesarios para la creación del laboratorio de pruebas.

Aunque en un principio consideramos que la mejor opción pasaba por la virtualización de todos los sistemas del laboratorio, en la práctica este no resultó ser una solución completamente adecuada, es por ello que se terminó utilizando la instalación de software directamente sobre el ordenador físico en el caso de Pfsense, y los Proxmox de virtualización.

La inclusión de una tarjeta de red, con cuatro bocas, prestada por parte de la Fundación I+D del Software Libre, ha sido crucial para la implementación de gestor de red de la simulación, sin esto no habría sido posible la implementación del software de firewall directamente sobre la máquina física. De hecho se comenzó mediante la virtualización del mismo, pero la generación de redes virtuales dentro de una red global, complicaba la interconexión con las máquinas que se encontraban en otros servidores, puesto que debían encontrarse en redes independientes y hubiese sido necesaria la generación de VLANs, para lo cual era necesario un Switch con dicha capacidad del cual no disponemos.

La interfaz WAN de nuestro ordenador, se conecta al router de telefónica, y se considera como una red externa, es decir la red 192.168.1.0/24 es la red desde la que se llevo a cabo la prueba de conexión VPN puesto que a todos los efectos es una red externa a la empresa, podría considerarse como una IP de otra empresa conectada al mismo proveedor de servicios que la nuestra. y el gateway de la casa, 192.168.1.1 es un gateway transparente para toda nuestra infraestructura, excepto para nuestro Firewall que es el único que conoce de su existencia.

Una vez tomada la decisión de usar discos duros limpios e instalar físicamente sobre las máquinas, el proceso se simplificó bastante. Tres ordenadores, cada uno conectado a una toma de la tarjeta de red del ordenador que hacía de Firewall/Enrutador y ya estaba la infraestructura física de red conformada, toda la parte lógica se definió dentro del Pfsense y bastaba con asignar ips de cada una de las redes definidas en los anfitriones de proxmox del BACKEND y del DMZ. La cuarta boca de red, para la red donde se ubicarían los ordenadores de desarrollo se activa con DHCP en el servidor Pfsense y ya desde un ordenador conectados a esa toma podemos realizar toda la configuración necesaria.

Se procede a instalar los Proxmox en los anfitriones y la Debian 7 Desktop en el portátil y una vez instalados se utilizará dicho portátil para acceder mediante interfaz web o SSH a los anfitriones de proxmox y terminar de configurarlos y levantar las máquinas virtuales.

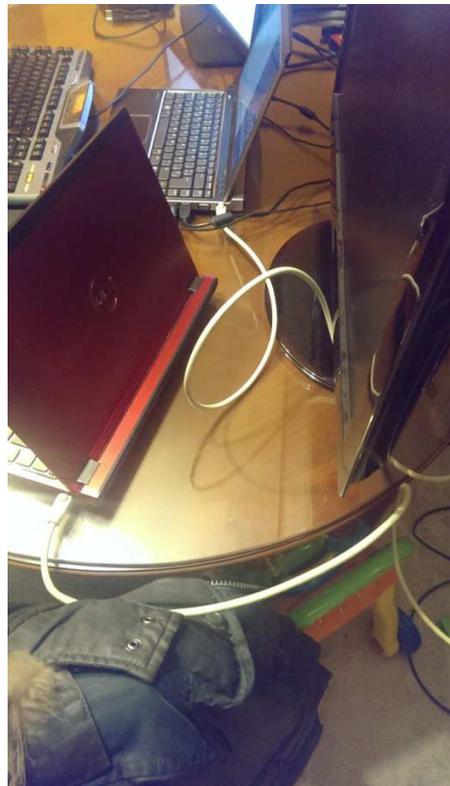
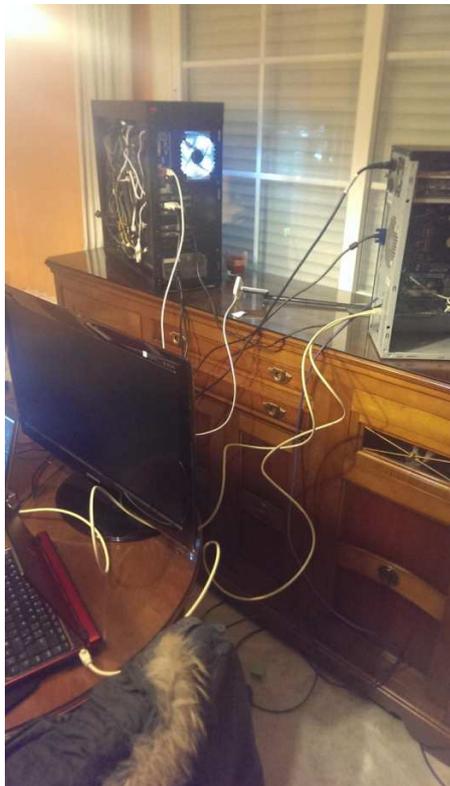
En este momento se desconectan las pantallas de dichos anfitriones y del Firewall puesto que en principio dejan de ser necesarias para continuar con la simulación y se conectan al ordenador de desarrollo para permitirnos agilizar toda la implementación.

Nos descargamos una debian base Netinstall en nuestro ordenador de desarrollo y ésta la empleamos para subirla a través de la interfaz web al los anfitriones de proxmox y con ella generar una plantilla de Debian 7 básica que será la que utilizemos de ahora en adelante para clonar y generar las diferentes máquinas.

En esta plantilla se incluyen herramientas básicas, como pueden ser el editor VIM, y se actualizan los repositorios a la última versión.

Entonces se empezamos a crear clones de las máquinas que al arrancar se cambian el nombre, así como la dirección IP y el archivo hosts. De este modo generamos todas las máquinas virtuales con las que hemos estado trabajando.

En las siguientes imágenes se puede observar con más detalle como hemos montado nuestro laboratorio de pruebas y como se ha interconectado:



La utilización de tanto hardware ha sido para procurar en la medida de lo posible simular el entorno de la mejor forma posible, para que no quedasen dudas en cuanto a la posibilidad de implementación en la realidad de toda la infraestructura planteada.

5. Conclusiones y resultados

5.1 Evaluación global del funcionamiento

Tras haber completado la simulación y la implementación del sistema, podemos concluir que la simulación ha funcionado a la perfección. Todos los servicios han estado operativos durante muchos días de forma continuada, siendo incluso accedido en ocasiones desde el exterior a través de la VPN configurada (mediante un NAT en el router externo al sistema de dicho puerto) y no ha planteado ningún problema de rendimiento, es decir todo el sistema ha funcionado según lo esperado.

El rendimiento de las máquinas ha estado por siempre por encima de lo requerido, puesto que no se ha generado la carga necesaria en ninguna de las máquinas como para necesitar más hardware. La única máquina que requirió un aumento de sus recursos fue la destinada a la instalación de Zimbra, el sistema de correo, puesto que al hacer uso de muy diversas tecnologías, la instalación no era capaz de completarse con sólo 512Mb de RAM que eran como estaban configuradas los clones originales.

La integración de todos los sistemas ha sido muy buena, debido al uso de tecnologías de software libre en todos ellos, así como las garantías de estabilidad presentadas por el sistema en su conjunto. Sin que en ningún caso se produjesen caídas inesperadas de servicios o falta de disponibilidad de los mismos.

Finalmente podemos concluir que el el objetivo fundamental de este trabajo se ha logrado con creces, puesto que hemos demostrado que una empresa de desarrollo de software es capaz de mostrarse completamente operativa gracias a la utilización íntegra en todos sus sistemas de GNU/Linux. Éste sienta las bases con las que construir un entorno eficaz, y productivo capaz de generar desarrollos hábiles pues

5.2 Ventajas del uso de GNU/Linux

La utilización de GNU/Linux, ha supuesto una ventaja en todos los sentidos. Partiendo de la base de utilizando software propietarios hubiese sido imposible la implementación de todos estos sistemas debido al precio que muchos de los recursos empleados tienen en sus versiones propietarias, y terminando con que en muchos casos la facilidad de implementación con la que nos encontramos en sistemas GNU/Linux es muy superior a la ofrecida por otros sistemas, por poner un ejemplo podríamos señalar la facilidad con la que hemos implementado un servidor

web o un directorio de usuarios frente a la complejidad de implementar uno bajo un windows Server, el cual incluye mucha más configuración de roles y servicios, prácticamente desde la instalación del sistema.

En nuestro caso hemos visto cómo partiendo de la misma base, un sistema operativo, con las herramientas basem hemos podido configurar desde un entorno de desarrollo para el usuario final, hasta el servidor de bases de datos que utiliza la empresa. Esto en un sistema propietario como es windows es imposible, puesto que podemos emplear una sola versión de windows para los portátiles de los usuarios y para los servidores de base de datos.

Otra de las ventajas detectadas durante la implementación de esta práctica es la eficiencia de recursos con la que trabajan los sistemas basados en GNU/Linux, tenemos el ejemplo de servidores como puede ser el servidor WEB al que sólo le hemos instalado Apache, y podemos ver como la utilización de recursos que hace es mínima:

```
root@web-pfc-uoc:~# free -m
              total        used         free       shared    buffers     cached
Mem:           497           91          405           0           28           27
-/+ buffers/cache:
Swap:        1020            0         1020
root@web-pfc-uoc:~#
```

El servidor en funcionamiento con el servidor web levantado y el servidor SSH, está ocupando 91MB de memoria en la máquina, dejando libres 405, esto es una cifra inalcanzable mediante la utilización de un sistema como windows, que prácticamente es incapaz de arrancar el sistema solo si no tiene como mínimo 512MB de ram, y eso sólo para iniciar el proceso.

Y la ventaja fundamental, es el coste, durante el desarrollo del trabajo, hemos empleado tecnologías y soluciones, que se encuentran al nivel de otras mucho más costosas basadas en software propietario. Por poner un ejemplo Proxmox, es un sistema de virtualización que poco o nada tiene que envidiarle a VMware o Parallels, sin embargo el coste de las licencias de los otros sistemas hacen que una PYME tenga una barrera en entrada muchas veces superior a lo que puede permitirse, lo cual fomenta en muchas ocasiones la piratería. Por mucho menos de lo que se ha de invertir en una licencia de cualquiera de esas soluciones propietarias, se puede contratar un soporte profesional por parte de Proxmox, que nos asegura que en caso de que tengamos algún problema con nuestro software de virtualización hay una empresa respaldando y que nos va a ayudar a superar ese problema.

Y por último no está de más comentar el hecho de la seguridad, al haber estado implementando toda la infraestructura con GNU/Linux y sistemas libres, podemos tener la seguridad de que nadie está obteniendo información de nuestros sistemas, y de que no se están produciendo envíos de información de ningún tipo hacia

los servidores de empresas de terceros que puedan haber incluido algún código cerrado en sus aplicaciones, de hecho es bastante impresionante el observar como el tráfico que genera un servidor de virtualización que mantiene 9 servidores virtualizados con servicios instalados en el mismo, se mantiene completamente a 0 mientras no se hace ningún trabajo o solicitud al mismo o a alguno de los servidores instalados, lo cual no ocurre cuando estamos empleando sistemas propietarios que por lo general envían todo tipo de información de uso y estadístico a los servidores centrales de los desarrolladores, y que lo más probable es que estemos aceptando en el momento que marcamos la casilla correspondiente en el EULA.

5.2 Posibles problemas de migración desde software propietario

Uno de los principales problemas cuando se plantea la migración desde un software que ya se está utilizando, es la resistencia de las personas en sí a cambiar su forma de trabajar. Cuando nos acostumbramos a emplear un determinado programa para llevar a cabo una tarea, hay poco interés en adaptarse a una nueva forma aunque esta sea más eficiente o más cómoda, esto es debido al esfuerzo que hay que realizar para adaptarse a esta nueva forma de trabajo. Por ello es importante que cualquier empresa que esté empleando soluciones de software propietario y se plantee realizar el cambio por soluciones en software libre, se plantee la realización de una campaña formativa que comunique a todos los usuarios los motivos por los que se produce el cambio y cuales son las ventajas de la implantación de esa nueva tecnología en la labor diaria.

Durante el desarrollo de este proyecto hemos conocido diversas soluciones que presentan un gran ahorro económico con respecto a soluciones privativas que en muchas ocasiones tienen las mismas e incluso menos funcionalidades, sería un verdadero error la no adopción o implantación de dichas tecnologías por el simple hecho de no superar esta barrera de resistencia al cambio que existe en todas las entidades.

La falta de confianza en el software libre es otro de los principales motivos que impiden el cambio en muchas ocasiones. Muchas de las herramientas de software libre están creadas y mantenidas por comunidades de personas en lugar de tener detrás a una empresa como tal, esto plantea muchas dudas y desconfianza sobre todo en los niveles de dirección de la empresa, en los que no se tiene un conocimiento profundo sobre cómo funcionan estas comunidades y cómo nos pueden ayudar a resolver cualquier problema que tengamos con dicho software. Muchas veces se asocia el pagar por un producto con la calidad del mismo cuando en la mayoría de las ocasiones no es así. Sin embargo lo que si hacen en muchas ocasiones estas empresas que nos venden el software privativo es generar más dependencia del mismo, mediante mecanismos como incompatibilidad de formatos en los archivos que generamos y demás inducen a la compra de otras soluciones de su mismo entorno que generan más dependencia aún si cabe.

Por ello es importante una sesión de información sobre software libre, en la que en todos los estamentos de la empresa participen y terminen siendo conscientes de cuáles son las principales ventajas de migración, puesto que hay muchas y más importantes de lo que pueden parecer en un principio.

Además estos temores en muchas ocasiones son infundados, puesto que siempre se pueden contratar servicios de muchas empresas que se dedican en muchas ocasiones a la implantación de software libre y a ofrecer soporte técnico y a un coste mucho menor que el coste de las licencias que pagamos por sistemas privativos que vienen en muchas ocasiones sin soporte de ningún tipo.

Por último cabe destacar que nos encontraremos con ocasiones en los que la alternativa libre a una solución propietaria puede no tener toda la funcionalidad que nos ofrece dicha solución. En ese caso debemos plantearnos si realmente hacemos un uso de dicha funcionalidad tan crítico como para que no puede ser sustituida por alguna otra solución, puesto que muchas veces utilizamos soluciones que ofrecen multitud de características que estamos utilizando en nuestro trabajo diario. Incluso es posible que nos encontremos con que determinadas soluciones con las que hemos estado trabajando han estado haciendo uso de formatos propietarios para almacenar los resultados y que el paso de esos datos en formatos propietarios a un formato libre tiene un coste que la empresa debe estar dispuesta a asumir y que se debe calcular en la estimación de riesgos de la migración.

En general hemos planteado algunos de los grandes inconvenientes de la migración hacia tecnologías libres en el entorno empresarial, pero como hemos visto a lo largo del desarrollo de este trabajo, el software libre nos ofrece unas soluciones muy completas y que en muchas ocasiones parecen más complicadas de manejar cuando no se tiene ningún conocimiento. Casi todas como hemos visto tienen interfaces de configuración y mantenimiento que distan mucho de las originales y costosas labores de trabajo mediante línea de comandos, la cual prácticamente queda reducida al momento de la instalación de las mismas, para luego ser mantenida via web o mediante entornos de ventana.

5.3 Sistemas alternativos

Durante el desarrollo de este trabajo se han valorado muchos otros sistemas que realizan asdla misma función que los sistemas escogidos, esto nos demuestra el gran abanico de posibilidades que nos ofrece el software libre a la hora de implementar los sistemas de nuestro entorno. Para demostrarlo vamos a nombrar algunos de estos sistemas valorados de forma que queden patente su existencia y quede a disposición del lector la decisión de optar por uno u otro en caso de querer implementar algún subsistema de dicho tipo:

- GNU/Linux:
 - Centos: <http://www.centos.org/>
 - Ubuntu Server: www.ubuntu.com/download/server
 - Gentoo: <http://www.gentoo.org/>

- Virtualización:
 - VirtualBox: <https://www.virtualbox.org/>
 - Xen: <http://www.xenproject.org/>

- Seguridad:
 - IpCop: <http://ipcop.org/>
 - F1 mail gateway: <http://www.scrolloutf1.com/>

- Monitorización:
 - PandoraFMS: <http://pandorafms.com/>
 - Icinga: <https://www.icinga.org/>

- Comunicaciones:
 - iRedmail: <http://www.iredmail.org/>
 - RoundCube: <http://roundcube.net/>
 - eJabberd: <http://www.ejabberd.im/>
 - Tigase: <http://www.tigase.org/>

- Gestión de proyectos:
 - Mantis Bugtracker: <http://www.mantisbt.org/>
 - Project-open: <http://www.project-open.org/>
 - dotProject: <http://www.dotproject.net/>

6. Anexos

6.1 Documentos y publicaciones de apoyo

Implementación de GNU/Linux para una empresa de desarrollo de software
David Manzano Fiestas

Guía de Referencia y aprendizaje Linux. Dlaheimer y Welsh

Editorial O'Reilly. Segunda edición.

Administración de sistemas Linux. Tom Adelstein y Bill Lubanovic.

Editorial O'Reilly

The Debian Administrator's Handbook. Raphaël Hertzog y Roland Mas.

<http://debian-handbook.info/browse/stable/>

6.2 Direcciones web relacionadas

GNU/Linux

<http://en.wikipedia.org/wiki/Linux>

http://es.wikipedia.org/wiki/Debian_GNU/Hurd

<http://es.wikipedia.org/wiki/GNU/Linux>

<http://www.debian.org/doc/manuals/debian-tutorial/ch-about.html>

<https://www.gnu.org/gnu/linux-and-gnu.html>

<https://www.gnu.org/gnu/the-gnu-project.html>

<https://www.gnu.org/gnu/gnu-linux-faq.html>

Empresa y Software Libre

http://en.wikipedia.org/wiki/Linux_adoption

<http://www.linax.es/index.php/linux-en-la-empresa>

<http://www.cloudtweaks.com/2012/08/advantages-and-disadvantages-of-open-source/>

http://www.pcworld.com/article/209891/10_reasons_open_source_is_good_for_business.html

Distribuciones Linux

<http://www.linux-es.org/distribuciones>

Virtualización

<http://www.vmware.com/virtualization/>

<http://www.webopedia.com/TERM/V/virtualiz>

<http://www.thecoffemaker.com.ar/2012/02/21/virtualizando-con-openvz/>

<http://www.linuxjournal.com/article/9764>

<http://www.proxmox.com/proxmox-ve>

Seguridad

<http://www.pfsense.org/>

<http://www.franciscosepulveda.eu/2013/05/21/monitorizacion-de-redes-con-nagios/>

Control de usuarios

<http://www.openldap.org/>

Comunicaciones

<http://www.versity.com/index.php/productos/zimbra/86.html>

<http://lifelifehacker.com/5940565/why-you-should-start-using-a-vpn-and-how-to-choose-the-best-one-for-your-needs>

Bases de datos

<http://dev.mysql.com/doc/refman/5.0/es/features.html>

http://danielpecos.com/docs/mysql_postgres/x15.html

Gestión de proyectos

<http://www.techopedia.com/definition/24735/netbeans>

http://www.ecured.cu/index.php/Eclipse,_entorno_de_desarrollo_integrado

<http://subversion.apache.org/features.html>

Otros

<http://es.libreoffice.org/caracteristicas/>