Universidad Carlos III de Madrid
Escuela Politécnica Superior
Departamento de Informática

Proyecto fin de carrera de Ingeniería Técnica en Informática de Gestión

Análisis, diseño e implantación de un sistema de servidores departamental basado en máquinas virtuales

Autor: **Francisco Olcina Grande**
Tutor: **Alejandro Calderón Mateos**

Leganés, Septiembre de 2007

A mi familia y a mis amig@s
que siempre han estado
ahí para mi.

"No hay mejor remedio para el alma que
los sentidos, igual que no hay mejor
remedio para los sentidos que el alma"
El retrato de Dorian Gray - Oscar Wilde.

Agradecimientos

Sin duda alguna, estos años que he pasado en la Universidad forman parte de una de las mejores etapas de mi vida, y éste proyecto pone el sello final a una ilusión que comenzó hace muchos años, quizá incluso antes de que yo naciese, dado que mis padres deseaban que sus futuros hijos tuviesen la educación de la que ellos no pudieron disfrutar. Les agradezco a ellos el haber podido llegar a donde he llegado, les doy las gracias por todo lo que han luchado por mi hermana y por mi, y también le doy las gracias a mi hermana por tener siempre su apoyo cuando lo he necesitado. Esta carrera no solo la he terminado yo, vosotros también junto a mi.

También quiero agradecer el haber podido finalizar estos estudios a toda la gente que de alguna manera, me ha ayudado, ya sea personalmente o profesionalmente. Espero acordarme de todos.

A mi profesor de filosofía del instituto, que me dijo que la próxima vez que me viese después de terminar COU esperaba que fuese conduciendo un BMW con una rubia de acompañante y un titulo de ingeniero bajo el brazo. (De momento voy a tener el titulo...jeje)

A los compañeros que tuve el primer año de ésta carrera y que aun sigo viendo: a Pili por estar siempre dispuesta a tomarse unas cervezas, a Alfon por los vicios que nos echábamos en el ciber en horas de clase, y a Adi por enseñarme nuevas recetas con nata.

A los técnicos del laboratorio de informática de la Universidad Oscar y Roberto, que fueron mis jefes cuando fui becario allí y que siempre han creído en mi.

A la gente de ARCOS: a Jesús por darme la oportunidad de seguir trabajando en la universidad, a David por ser un buen compañero de carreras y de piques en los entrenamientos, a Felix por su opinión sobre cierta web en inglés y apoyo a otra mia, a Javi (Doc) por ayudarme tanto con el proyecto de EADS, a José Daniel por sus consejos de Latex, a Florin por ser un maestro en el arte del ligoteo en estaciones de ski, a Juan Carlos (Pichel) por sus consejos sobre literatura y cine checo, a Rosa por ser la que más nos cuida en ARCOS, a Cana por introducirme en el Snowboard y el Bodyboard, a Alejandra por querer compartir conmigo la emoción de un buen salto de Snow, a Javi (Zor) y a su novia Penélope, por ser tan atentos y buenas personas, a Lusimi por nuestras conversaciones de rock clásico, a Sole por su simpatía y conversaciones sobre libros, a Fortran por ser tan autentico y dejar meterle mano a su ...coche, a Borja por toda la información para ver canales de satélite de forma más económica, a Laura por las conversaciones profundas sobre la vida que surgían en la fiesta de Caipirinha de Florin, a Marga por ayudarme

siempre que lo he necesitado, a Fer por las partidas a Frets on Fire en el laboratorio, y sobre todo a mi tutor Alex, por apoyarme de principio a fin y creer en mi.

A otros compañero de trabajo, pero de la Peugeot: a Juan, por ser un gran compañero con quien pasar las noches de curro hablando de cualquier cosa, a Dani por haberme enseñado tanto sobre la vida, a Carlos (cajones) por ser otro gran compañero y un gran amigo, y a Antonio (cajones) por las risas que nos echamos los viernes noche.

Y por supuesto no podían faltar mis amig@s, que siempre han estado ahí y me han dado momentos de felicidad de los que siempre les estaré agradecido: a Míguel y Mónica, la eterna pareja con quien paso casi todos mis findes y a los que quiero muchísimo, a Felipe por ser un gran amigo y por haber vivido juntos tantas cosas y las que nos quedan por vivir, a Tania (su novia), por escucharme y tener siempre una sonrisa que ofrecer, a Elena por ser la mejor compañera de prácticas que he tenido nunca y ser como es conmigo, a Rober por demostrarme lo que son 7 pintas, a Juan (banano), Tito, Miguel (Dopacio), Gomez, Dani y Denis por todas las risas que hemos pasado juntos y que me han hecho feliz, a Natalia por su gran empatía conmigo, a Eva por descubrirme el Jägermeister, a Laura (Sweetlaura) por cuestiones licántropas, y a Ana por ser una de las mejores personas que he conocido y a la que quiero muchísimo.

Francisco Olcina Grande,
Paco, para todos vosotros.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

En este capítulo se hace una breve introducción del proyecto indicando el origen y los objetivos, además de mostrar la estructura de éste documento.

1.1. Origen

Este proyecto se ha realizado en el grupo *ARCOS* del área de Arquitectura y Tecnología de Computadores, en el cual he trabajado en calidad de becario.
El origen del mismo parte de las limitaciones existentes en el antiguo sistema de servidores del grupo, y de la idea de utilizar máquinas más modernas para hacer una nueva implantación de un sistema de servidores. Se necesitaba un diseño más innovador de cara a introducir las nuevas capacidades que se habían planteado, como aumentar la capacidad de disco, la fiabilidad de los datos y la seguridad de todo el sistema.

La disposición física del antiguo sistema de servidores era poco adecuada: todas las máquinas que componían el sistema estaban dentro del laboratorio, bajo una temperatura inadecuada y con un número de sistemas *SAI* insuficiente para todas las máquinas. La Universidad permitió al grupo *ARCOS* introducir un armario con máquinas en forma de *rack* en el centro de cálculo, contando de esta manera con una temperatura estable y adecuada, así como un sistema *SAI* permanente para todas las máquinas integradas en el armario.

Para llevar a cabo la implantación del nuevo sistema de servidores, se compró una máquina de gran capacidad de almacenamiento y potencia de cómputo con las siguientes características:

- *Intel Core 2 DUO 2,13Ghz* con $4GB$ de memoria *RAM*. Viene en formato de *rack* con dimensión 4U(ocupa 4 huecos en el armario). Dispone de 8 discos duros *Serial*

ATA con distintas capacidades de almacenamiento.

También se utilizó una de las máquinas destinadas a formar un cluster de cómputo, con las siguientes características:

- *Intel Pentium 4 2.8Ghz* con $1GB$ de memoria *RAM*. Viene en formato de *rack* con dimensión 3U (ocupa 3 huecos en el armario). Dispone de 3 discos duros de $250GB$ de capacidad.

Por último se reutilizó una de las máquinas de componía el antiguo sistema de servidores. Ésta máquina tiene las siguientes características:

- *AMD K7 ATHLON XP 2000+* con $512MB$ de memoria *RAM*. Viene en formato *ATX* normal, no tiene formato de *rack* y dispone de 4 discos duros *ATA* de $120GB$ de capacidad.

La idea era aprovechar las 2 máquinas en formato *rack* para que formasen parte del armario que *ARCOS* iba a establecer en el centro de cálculo de la Universidad. La tercera máquina se quedaría en el laboratorio, ofreciendo una funcionalidad de respaldo de datos aprovechando la distancia física, como se verá mas adelante en el presente documento.

1.2. Objetivo

El objetivo de éste proyecto ha sido llevar a cabo un análisis, un diseño y una implantación de un nuevo sistema de servidores para el grupo *ARCOS*. Éste sistema ha seguido un enfoque que ofrece fiabilidad, rapidez, disponibilidad y seguridad.

Para llevar a cabo el objetivo se han utilizado las máquinas listadas anteriormente y se han contemplado las soluciones software existentes en materia de servidores, desde los sistemas operativos hasta el software utilizado para los servicios a ofrecer.

El nuevo sistema debe ofrecer una mejora de los servicios que se ofrecían al personal de ARCOS, mediante el aprovechamiento de las nuevas máquinas disponibles para implantarlo. Éstos servicios son principalmente los siguientes:

- Almacenamiento de datos.
- Cuenta de correo.
- Disponibilidad de tener páginas web.
- Acceso por ssh a una máquina Linux.
- Servicio de resolución de nombres (DNS).

- Servicio de DHCP.
- Autenticación en las máquinas con Windows y Linux del laboratorio.

1.3. Estructura del documento

El documento se divide en los siguientes capítulos:

- **Introducción**, donde se exponen el origen y los objetivos del proyecto, además de la estructura del presente documento.
- **Estado de la cuestión**, donde se expone toda la información relativa a las infraestructuras hardware y software existentes para la solución que se quiere alcanzar. La información se divide en varios apartados: sistemas operativos, máquinas virtuales, sistemas de autenticación, otros servicios y por último, aumento de fiabilidad de datos.
- **Análisis**, donde se muestra el análisis realizado.
- **Diseño**, donde se muestra el diseño realizado.
- **Implantación**, donde se muestran las fases que se han llevado a cabo para la implantación del nuevo sistema de servidores.
- **Resultados**, donde se muestran las resultados obtenidos en el sistema, y se sacan conclusiones de los mismos.
- **Conclusiones y trabajos futuros**, donde se exponen las conclusiones extraídas después del trabajo realizado y en base al funcionamiento del sistema obtenido. Además se incluye un presupuesto, y se explican los futuros trabajos de mejora y ampliación del sistema.
- **Bibliografía**, donde se muestran las fuentes externas de información que han sido necesarias para la realización del proyecto y del presente documento.
- **Apéndices**, donde se muestra la instalación de diversas máquinas, así como ficheros de configuración.

Capítulo 2

Estado de la cuestión

Aquí se analizan las diferentes infraestructuras necesarias para la creación del sistema de servidores de este proyecto.

2.1. Sistemas operativos

Un sistema operativo (SO) es un conjunto de programas destinados a permitir la comunicación del usuario con un ordenador y gestionar sus recursos de manera eficiente. Comienza a trabajar cuando se enciende el ordenador, y gestiona el hardware de la máquina desde los niveles más básicos.

Un sistema operativo se puede encontrar normalmente en la mayoría de los aparatos electrónicos que podamos utilizar sin necesidad de estar conectados a un ordenador y que utilicen microprocesadores para funcionar, ya que gracias a estos podemos entender la máquina y que ésta cumpla con sus funciones (teléfonos móviles, reproductores de DVD, autoradios... y computadoras)
Existen varias familias de sistemas operativos que se enumerarán y tratarán a continuación:

2.1.1. Microsoft Windows

Microsoft Windows es un sistema operativo gráfico para computadoras personales cuyo propietario es la empresa Microsoft. Desde sus inicios, han salido al mercado distintas versiones, pasándose a detallar a continuación las más actuales:

- **Microsoft Windows 2000**: es un sistema operativo basado en el núcleo de Windows NT; pertenece a la familia de servidores de Microsoft, proporcionando una gran cantidad de servicios en red e innovaciones tecnológicas respecto a sus prede-

cesores. En cuanto a sus características más destacadas:

  - Almacenamiento: tiene soporte para particiones de tipo FAT16, FAT32 y NTFS, servicio de indexación, encriptación de ficheros, sistema de archivos distribuido y permite la creación de los distintos tipos de *RAID* por software.
  - Comunicaciones: tiene soporte para acceso remoto mediante Terminal Server, balanceo de carga, servidor web y de correo (IIS), y directorio activo que almacena la información de los objetos del dominio.

- **Microsoft Windows XP**: La unión de Windows NT/2000 y la familia de Windows 9.x se alcanzó con Windows XP liberado en 2001 en su versión *Home* y *Professional*. Windows XP usa el núcleo de Windows NT. Incorpora una nueva interfaz y hace alarde de mayores capacidades multimedia. Además dispone de otras novedades como la multitarea mejorada, soporte para redes inalámbricas y asistencia remota. Dado que no pertenece a la familia de servidores de Microsoft, no puede actuar como un servidor de dominio, estando limitado a actuar como cliente del mismo.

- **Microsoft Windows Server 2003**: Sucesor de la familia de servidores de Microsoft a Windows 2000 Server. Es la versión de Windows para servidores lanzada por Microsoft en el año 2003. Está basada en el núcleo de Windows XP, al que se le han añadido una serie de servicios, y se le han bloqueado algunas de sus características (para mejorar el rendimiento, o simplemente porque no serán usadas).

Las principales ventajas y desventajas de los sistemas operativos de Microsoft son las siguientes:

Ventajas:

- Facilidad de uso al poder realizarse cualquier tarea usando el entorno gráfico.
- Aconsejable para entornos empresariales que no tengan a gente especialista en sistemas operativos para realizar el mantenimiento de un servidor o un equipo de escritorio que use Microsoft Windows.

Desventajas:

- No se puede acceder al código fuente.
- Parametrización muy limitada, poca flexibilidad del sistema operativo.
- Necesita de equipos más potentes y con más memoria, al tener un entorno gráfico obligatorio.

- Al ser un software propietario y de código cerrado, cualquier vulnerabilidad encontrada, debe ser resuelta por el personal de Microsoft, y si esta resolución se considera oportuna. Si se pudiese acceder al código, se encontrarían vulnerabilidades que rápidamente serían resueltas por el resto de la comunidad informática, no dejando ninguna sin resolver.

2.1.2. Unix

UNIX es un sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

Familias

Existen varias familias del sistema operativo UNIX que han evolucionado de manera independiente a lo largo de los años. Cada familia se distingue no tanto por sus diferencias técnicas como por sus diferencias en propiedad intelectual. Se observa que todas las familias se han visto contaminadas, directa o indirectamente, por otras familias. Las familias UNIX más significativas son:

- **AT&T**: la familia que tuvo su origen en el UNIX de AT&T. Considerada la familia UNIX "pura" y original. Sus sistemas operativos más significativos son UNIX System III y UNIX System V.
- **BSD**: familia originada por el licenciamiento de UNIX a Berkely. BSD incorpora propiedad intelectual no originaria de AT&T, la primera implementación de los protocolos TCP/IP que dieron origen a Internet.
- **AIX**: esta familia surge por el licenciamiento de UNIX System III a IBM.
- **Xenix**: familia derivada de la adquisición de los derechos originales de AT&T por parte de SCO.
- **GNU/Linux**: En 1983, Richard Stallman anunció el Proyecto GNU, un ambicioso esfuerzo para crear un sistema similar a Unix, que pudiese ser distribuido libremente. El software desarrollado por este proyecto -por ejemplo, GNU Emacs y GCC - también han sido parte fundamental de otros sistemas UNIX. En 1991, cuando Linus Torvalds empezó a proponer el *kernel* Linux y a reunir colaboradores, las herramientas GNU eran la elección perfecta. Al combinarse ambos elementos, conformaron la base del sistema operativo (basado en POSIX) que hoy conocemos como GNU/Linux o simplemente Linux. Las distribuciones basadas en el *kernel*, el software GNU y otros agregados entre las que podemos mencionar a Red Hat Linux y Debian GNU/Linux se han hecho populares tanto entre los aficionados a la

computación como en el mundo empresarial. Obsérvese que Linux tiene un origen independiente, por lo que se considera un 'clónico' de UNIX y no un UNIX en el sentido histórico.

Implementaciones más importantes

A lo largo de la historia ha surgido una gran multitud de implementaciones comerciales de UNIX. Sin embargo, un conjunto reducido de productos han consolidado el mercado y prevalecen gracias a un continuo esfuerzo de desarrollo por parte de sus fabricantes. Los más importantes son:

- **Solaris de Sun Microsystems**. Uno de los sistemas operativos Unix más difundido en el entorno empresarial y conocido por su gran estabilidad. Parte del código fuente de Solaris se ha liberado con licencia de fuentes abiertas.
- **AIX de IBM**. Históricamente, IBM ha potenciado sus sistemas operativos de mainframe (tales como OS/390) y consideró UNIX como una mera curiosidad sin futuro. Cuando las computadoras departamentales basadas en UNIX (entonces denominadas minicomputadores o minis) empezaron a proliferar, IBM decidió no perder mercado introduciéndose en este segmento. Actualmente, IBM ha abandonado el desarrollo de todos sus sistemas operativos para centrarse en Linux.
- **HP-UX de Hewlett-Packard**. Este sistema operativo también nació ligado a las computadoras departamentales de este fabricante. También es un sistema operativo estable que continua en desarrollo.

Las principales ventajas y desventajas de los sistemas operativos UNIX son las siguientes:

Ventajas

- Puede ofrecer la funcionalidad de servidor sin necesitar interfaces gráficos, aumentando el rendimiento al no necesitar más CPU y memoria para un interfaz gráfico.
- Se trata de un sistema operativo muy consolidado en el mundo empresarial para su uso en servidores, debido a su antiguedad y por tanto, mejora a lo largo de los años.
- Soporta múltiples arquitecturas.

Desventajas

- Se trata generalmente de un sistema operativo de código cerrado, salvo OpenSolaris y algunas derivaciones de Unix como ciertos BSD. (Linux no se considera un Unix estrictamente hablando).
- El coste por licencia y mantenimiento suele ser elevado, al tratarse de productos enfocados a empresas.

2.1.3. Linux

Linux es la denominación de un sistema operativo y el nombre de un núcleo. Es uno de los paradigmas del desarrollo de software libre (y de código abierto), donde el código fuente está disponible públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo.

El término Linux estrictamente se refiere al núcleo Linux, pero es más comúnmente utilizado para describir al sistema operativo tipo Unix (que implementa el estándar (POSIX), que utiliza primordialmente filosofía y metodologías libres (también conocido como GNU/Linux) y que está formado mediante la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software (libre o no libre). El núcleo no es parte oficial del proyecto GNU (el cual posee su propio núcleo en desarrollo, llamado (Hurd), pero es distribuido bajo los términos de la licencia GNU GPL.

La expresión Linux también es utilizada para referirse a las distribuciones GNU/Linux, colecciones de software que suelen contener grandes cantidades de paquetes además del núcleo. El software que suelen incluir consta de una enorme variedad de aplicaciones, como: entornos gráficos, suites ofimáticas, servidores web, servidores de correo, servidores FTP, etcétera. Coloquialmente se aplica el término Linux a éstas, aunque en estricto rigor sea incorrecto, dado que la distribución es la forma más simple y popular para obtener un sistema GNU/Linux.

Las principales ventajas y desventajas de los sistemas operativos GNU/LINUX son las siguientes:

Ventajas:

- Puede ofrecer la funcionalidad de servidor sin necesitar interfaces gráficos, aumentando el rendimiento al no necesitar más CPU y memoria para un interfaz gráfico.
- Soporta múltiples arquitecturas.
- Al tratarse de un software de código abierto, la comunidad informática contribuye a su desarrollo y mejora, produciendo un sistema operativo concorde a las tecnologías y necesidades actuales.
- Existen múltiples herramientas producidas, también de código abierto, que pueden ejecutarse desde Linux.

Desventajas:

- Para usuarios poco avanzados, puede ser más costosa su utilización, dado que no existen herramientas gráficas que faciliten todas las operaciones que se pueden realizar en el sistema operativo.
- Incompatibilidad con las aplicaciones creadas para Windows, salvo que se utilicen emuladores del mismo.

A continuación se muestra una comparativa entre las distribuciones más importantes:

	Kernel	**Sistema de ficheros**	**Sistema de ficheros soportado**	**Arquitectura**
Debian GNU/Linux	Linux 2.4.27/2.6.18	Ext3	ext2, JFS, XFS, FAT, NTFS, ISO 9660, UDF, NFS, ReiserFS	x86, x86-64, IA64, PPC, SPARC, SPARC64, Alpha, MIPS, ARM, PA-RISC, Mac/VME 68k, S/390
Fedora Core	Linux 2.6.18 Fed.Core. Verid 9	Ext3	ext2, ReiserFS, FAT, ISO 9660, UDF, NFS	x86, x86-64, i386, PowerPC
Rxart	Linux 2.6.34	Ext3	ext2, ReiserFS, FAT, ISO 9660, UDF, NFS	x86, x86-64, i386, PowerPC
Mandriva Linux	Linux 2.6.12.12	Ext3	ext2, JFS, XFS, FAT, NTFS, ISO 9660, UDF, NFS, ReiserFS	x86 (i586), x86-64, PPC
Slackware Linux	Linux 2.4.33.3 / 2.6.18	ReiserFS, ext3/ext2	JFS, XFS, FAT, NTFS, ISO 9660, UDF, NFS	x86, IA64, S/390
SUSE Linux	Linux 2.6.11.4	ReiserFS	ext2, ext3, JFS, XFS, FAT, NTFS, ISO 9660, UDF, NFS, Reiser4	x86, IA64, x86-64, PPC

	Herramienta de actualización online	**Administrador de paquetes**	**Paquetes**
Debian GNU/Linux	APT	dpkg, Synaptic, APT, Adept y Aptitude	18000
Fedora Core	up2date, yum, APT (limitado)	RPM, yum	5000
Rxart	up2date, net, APT (limitado)	DEB, ASK	4000
Mandriva Linux	urpmi	RpmDrake	2000
Slackware Linux	Swaret, Slapt-get, y otras no oficiales	installpkg y upgradepkg	muchos
SUSE Linux	YaST2 (Misma Versión de YaST Avanzada)	RPM, YaST	12500

Cuadro 2.1: Distribuciones de Linux

2.1.4. Otros

Además de los sistemas operativos de la familia Windows, Unix y derivados, existen otros tales como Mac Os X, dirigidos principalmente a máquinas Apple con una arquitectura distinta de la *x86*.

2.2. Máquinas virtuales

Una máquina virtual es un software que crea un entorno virtual entre la plataforma de la computadora y el usuario final, permitiendo que este ejecute un software determinado.

El concepto de máquina virtual surge con el sistema VM/370 de IBM en 1972. La idea principal es la de permitir ejecutar varios sistemas operativos simultáneamente sobre el mismo hardware. Para ello, separa las dos funciones básicas que realiza un sistema de tiempo compartido: multiprogramación y abstracción del hardware.

El corazón del sistema es conocido como monitor de máquina virtual, y se ejecuta sobre el hardware proporcionando varias máquinas virtuales al siguiente nivel de software. Por eso cada una puede estar ejecutando un sistema operativo distinto.

Básicamente se pueden considerar 3 tipos de virtualización: emulación, virtualización completa (*Full Virtualization*), paravirtualización (*Paravirtualization*).

Emulación

La emulación se basa en crear máquinas virtuales que emulan el hardware de una o varias plataformas hardware distintas. Este tipo de virtualización es la más costosa y la menos eficiente, ya que obliga a simular completamente el comportamiento de la plataforma hardware a emular e implica también que cada instrucción que se ejecute en estas plataformas sea traducida al hardware real. Sin embargo la emulación tiene característi-

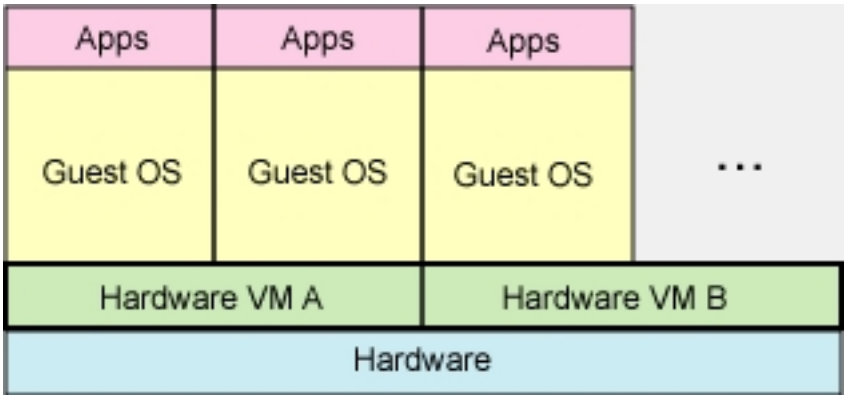


Figura 2.1: Emulación

cas interesantes, como poder ejecutar un sistema operativo diseñado para una plataforma concreta sobre otra plataforma, sin tener que modificarlo, o en el desarrollo de *firmware* para dispositivos hardware, donde se pueden comenzar estos desarrollos sin tener que esperar a tener disponible el hardware real.

Virtualización completa

Con este término se denominan aquellas soluciones que permiten ejecutar sistemas operativos huésped (*Guest*), sin tener que modificarlos, sobre un sistema anfitrión (*Host*), utilizando en medio un *Hypervisor* o *Virtual Machine Monitor* que permite compartir el hardware real. Esta capa intermedia es la encargada de monitorizar los sistemas huésped con el fin de capturar determinadas instrucciones protegidas de acceso al hardware, que no pueden realizar de forma nativa al no tener acceso directo a él.

Su principal ventaja es que los sistemas operativos pueden ejecutarse sin ninguna modificación sobre la plataforma, aunque como inconveniente frente a la emulación, el sistema operativo debe estar soportado en la arquitectura virtualizada. En lo que respecta al ren-

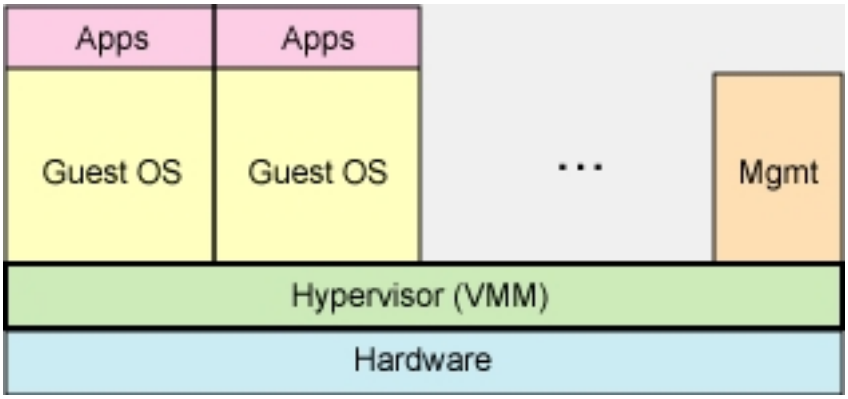


Figura 2.2: Virtualización completa

dimiento, éste es significativamente mayor que en la emulación, pero menor que en una plataforma nativa, debido a la monitorización y la mediación del *hypervisor*. Sin embargo, recientes incorporaciones técnicas en las plataformas *x86* hechas por Intel y AMD, como son Intel VT y AMD-V, han permitido que soluciones basadas en la virtualización completa se acerquen prácticamente al rendimiento nativo.

Hay que tener en cuenta también que la virtualización completa no se refiere a todo el conjunto de hardware disponible en un equipo, sino a sus componentes principales, básicamente el procesador y memoria. De esta forma, otros periféricos como tarjetas gráficas, de red o de sonido, no se virtualizan. Las máquinas huésped no disponen de los mismos

dispositivos que el anfitrión, sino de otros virtuales genéricos. Por ejemplo, si se dispone de una tarjeta nVidia GeForce en el anfitrión, los equipos huésped no verán esta tarjeta sino una genérica Cirrus.

Paravirtualización

La paravirtualización surgió como una forma de mejorar la eficiencia de las máquinas virtuales y acercarlo al rendimiento nativo. Para ello se basa en que los sistemas virtualizados (huésped) deben estar basados en sistemas operativos especialmente modificados para ejecutarse sobre un *hypervisor*. De esta forma no es necesario que éste monitorice todas las instrucciones, sino que los sistemas operativos huésped y anfitrión colaboran en la tarea. Uno de los componentes más destacados de esta familia es Xen. Permite para-

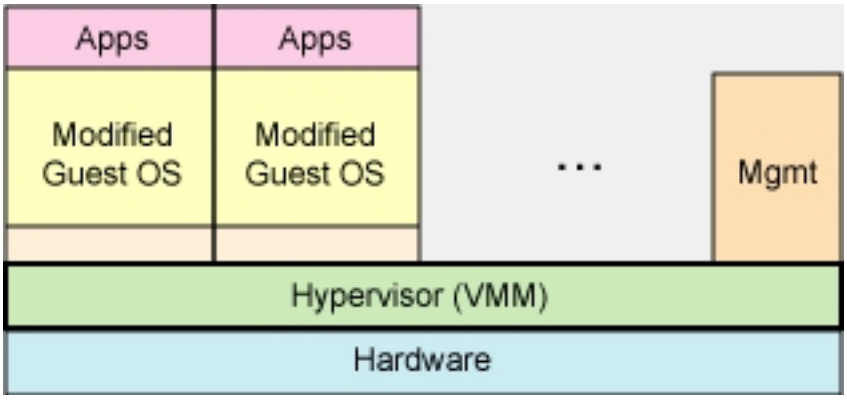


Figura 2.3: Paravirtualización

virtualización utilizando sistemas operativos modificados, y virtualización completa sobre procesadores con tecnología Intel-VT o AMD-V.

Usos de la virtualización

Los usos de la virtualización pueden ser muy variados, desde el simple echo de necesitar ejecutar una aplicación que no existe en tu sistema operativo hasta el testeo de sistemas operativos. Los más importantes usos son los siguientes:

- **Aprovechamiento de servidores**: los servidores salvo en casos excepcionales están infrautilizados, con la virtualización se pueden correr varios sobre la misma maquina y así aprovechar mejor las maquinas, reduciendo el espacio ocupado por los servidores y el consumo de energía.
- **Desarrollo**: se puede desarrollar una aplicación que deba de ejecutarse en distintas plataformas, como ejemplo se podría mencionar el desarrollo de sistemas operativos

o algo mucho mas común, el desarrollo de paginas web sobre Linux y su visualización sobre Internet Explorer.

- **Plataformas obsoletas**: a menudo se necesita usar aplicaciones antiguas que solo corren sobre un hardware antiguo, de esta manera se podría migrar esta maquina real a una virtual y dejar de depender del hardware.
- **Seguridad**: es frecuente separar los servicios ofrecidos en una red en distintas máquinas por razones de seguridad.

A continuación se muestran las distintas soluciones existentes para utilizar máquinas virtuales:

2.2.1. Linux-VServer

Linux-VServer es una implementación de servidor privado virtual que utiliza las capacidades de virtualización del Sistema Operativo y distribuida como software de código abierto, licenciada bajo GPL.

Linux-VServer es un mecanismo de jaula en el cual se pueden usar de manera segura los recursos de un sistema informático (tales como el sistema de archivos, tiempo de la CPU, direcciones de red y memoria) en tal manera que los procesos no pueden realizar un ataque de denegación de servicio sobre algo que estuviere por fuera de su partición.

A cada partición se le asigna un contexto de seguridad, y el sistema virtualizado dentro de aquella es el servidor privado virtual. Se proporciona un utilitario al estilo *chroot* para descender a los contextos de seguridad. Los contextos mismos son lo suficientemente robustos para arrancar muchas distribuciones de Linux sin modificar, incluyendo Debian y Fedora Core.

Los servidores privados virtuales se usan comúnmente en servicios de alojamiento web, donde son útiles para segregar cuentas de los clientes, agrupar recursos y contener cualquier brecha de seguridad potencial.

El Linux-Vserver no se incluye en la serie principal de desarrollo del núcleo. Linux-VServer 2.0, la versión estable hasta septiembre de 2005, existe como un parche para los núcleos de la serie 2.6. También se proporciona un parche para las series 2.4.

Ventajas:

- Muy rápidos y livianos, los servidores virtuales comparten la misma interfaz de llamada del sistema y no tienen ningún consumo por emulación.

- Los servidores virtuales pueden compartir un sistema de archivos común y no tienen que estar respaldados por imágenes opacas de discos.
- Los procesos dentro del servidor virtual se ejecutan como procesos regulares en el sistema anfitrión. Esto es algo más eficiente en cuanto a memoria y E/S se refiere que una emulación de un sistema completo, la cual no puede entregar memoria sin uso o compartir un caché de disco con el anfitrión.

Desventajas:

- Requiere parchear el núcleo del anfitrión.
- Todos los servidores virtuales comparten el mismo núcleo y por lo tanto se exponen a los mismos *bugs* y potenciales agujeros de seguridad.
- No se incluyen capacidades de cluster o de migración de procesos, de manera que el núcleo del anfitrión y la computadora anfitrión son un único punto de fallo para todos los servidores virtuales (similar a Xen y UML).
- La red no está completamente virtualizada (todavía) y los servidores virtuales son comúnmente meros aliases asignados de la misma interfaz de red. esto impide que cada servidor virtual cree su encaminamiento interno o configuración de cortafuegos propios.
- El límite de velocidad de *E/S* no se puede configurar por servidor virtual.
- Algunas llamadas del sistema (por ejemplo, aquellas que tratan con el reloj de tiempo real y las partes de los sistemas de archivos */proc* y */sys* permanecen sin implementar o sin virtualizar. Esto puede impedir que algunas distribuciones (especialmente Gentoo) arranquen apropiadamente dentro de un Linux-VServer sin modificaciones.
- No tiene soporte para IPv6.

2.2.2. QEmu

QEmu es un programa que ejecuta máquinas virtuales dentro de un sistema operativo, ya sea Linux, Windows, etc. Esta máquina virtual puede ejecutarse en cualquier tipo de Microprocesador o arquitectura (*x86*, *x86-64*, *PowerPC*, *MIPS*, *SPARC*, etc). Está licenciado en parte con la LGPL y la GPL de GNU.

El objetivo principal es emular un sistema operativo dentro de otro sin tener que hacer reparticionamiento del disco duro, empleando para su ubicación cualquier directorio dentro de éste.

El programa no dispone de interfaz gráfica, pero existe otro programa llamado QEmu Manager que hace las veces de interfaz gráfica si se utiliza QEmu desde Windows. También existe una versión para Linux llamado QEmu-Launcher En Mac OS X puede utilizarse el programa *Q* que dispone de una interfaz gráfica para crear y administrar las máquinas virtuales.

QEmu posee dos modos de operación: emulación del modo usuario, en la cual puede ejecutar procesos compilados para una determinada CPU en otra CPU, y emulación del sistema completo, incluyendo el procesador y varios periféricos. Además tiene implementado como módulo para el *kernel* Linux, un acelerador que aumenta la velocidad de emulación de *i386* en plataformas *i386* hasta un nivel ligeramente inferior a ejecutar en modo nativo. Se alcanza lo dicho ejecutando el modo de usuario y virtual en modo de código 8086 directamente sobre la CPU del computador. Además, sólo se usa la emulación del procesador y de los periféricos en modo *kernel* y en modo de código real.

Ventajas:

- Soporte para una gran variedad de arquitecturas, tanto en el *host*, como en la máquina emulada.
- Implementa el modo *Copy-On-Write* de formato de imagen de disco. Se puede declarar una imagen de disco de varios *gigas* y realmente ocupara en disco el tamaño que se este utilizando.
- El sistema *host* no ha de ser parcheado.
- Lleva integrado un servidor VNC para controlar remotamente las máquinas virtuales.

Desventajas:

- Soporte incompleto para Microsoft Windows y otros sistemas operativos anfitriones.
- Usado sobre arquitectura *x86* es menos eficiente que otras soluciones como VMWare, a menos que se utilice el acelerador, al estar basado en emulación.
- Más dificil de instalar y ejecutar que otras soluciones.

2.2.3. VMWare

VMware es un sistema de virtualización por software. VMware es similar a su homólogo Virtual PC, aunque existen diferencias entre ambos que afectan a la forma en la que el software interactúa con el sistema físico. El rendimiento del sistema virtual varía dependiendo de las características del sistema físico en el que se ejecute, y de los recursos

virtuales (CPU, RAM, etc) asignados al sistema virtual.

Mientras que VirtualPC emula una plataforma *x86*, Vmware la virtualiza, de forma que la mayor parte de las instrucciones en VMware se ejecutan directamente sobre el hardware físico, mientras que en el caso de Virtual PC se traducen en llamadas al sistema operativo que se ejecuta en el sistema físico.
VMware cuenta con varios productos:

- **VMware Player**: Es un producto gratuito que permite correr máquinas virtuales creadas con otros productos de VMware, pero no permite crearlas él mismo. Las máquinas virtuales se pueden crear con productos más avanzados como VMware Workstation.
- **VMware Server (antes GSX)**: En un principio era una versión de pago, hace unos meses fue liberada para ser descargada y utilizada de forma gratuita. Esta versión, a diferencia de la anterior, tiene un mejor manejo y administración de recursos; también corre dentro de un sistema operativo (*host*), está pensada para responder a una demanda mayor que el VMware Workstation.
- **VMware Workstation**: Es uno de los más utilizados pues permite la emulación en plataformas *x86*, esto permite que cualquier usuario con una computadora de escritorio o portátil pueda emular tantas máquinas virtuales como los recursos de hardware lo permitan. Esta versión es una aplicación que se instala dentro de un sistema operativo (*host*) como un programa estándar, de tal forma que las máquinas virtuales corren dentro de esta aplicación, existiendo un aprovechamiento restringido de recursos.
- **VMware ESX Server**: Esta versión es un sistema complejo de virtualización, pues corre como sistema operativo dedicado al manejo y administración de máquinas virtuales dado que no necesita un sistema operativo *host* sobre el cual sea necesario instalarlo. Pensado para la centralización y virtualización de servidores, esta versión no es compatible con una gran lista de hardware doméstico.

Funcionamiento: En el caso de la versión Workstation y Server, el funcionamiento es bastante similar a lo siguiente:
aplicación $-->$ *OS* (virtual) $-->$ Hardware (virtual) $-->$ VMware $-->$ (*OS host*) $-->$ hardware físico.
Esto afecta el rendimiento y desempeño de las máquinas virtuales, a diferencia de la versión ESX que funciona más o menos de la siguiente manera.

aplicación $-->$ *OS* (virtual) $-->$ Hardware (virtual) $-->$ VMware $-->$ hardware físico.

Ventajas:

- Facilidad de uso.
- Posibilidad de ejecutar imágenes de máquinas virtuales creadas en Virtual PC.
- Buen rendimiento obtenido mediante la técnica de virtualización..

Desventajas:

- Aunque se pueden utilizar los productos gratuitos de VMware para ejecutar máquinas virtuales y existen otros productos en el mercado para crearlas, si se quiere tener un rendimiento óptimo es necesaria una licencia para VMware ESX Server.
- La versión VMware ESX Server, que posee el rendimiento más óptimo de todas las soluciones VMware, no es compatible con una gran lista de hardware doméstico.

2.2.4. Microsoft Virtual PC

Programa desarrollado por Connectix y comprado por Microsoft para crear máquinas virtuales. Virtual PC, en el caso de la versión para Windows, no emula el procesador sino que deja que el mismo ejecute las instrucciones en el entorno emulado. Por el contrario, en la versión para MacOS emula un procesador Intel Pentium II. El resto del hardware que emula es: una placa con un chip Intel 440BX, una tarjeta de video S3 Tríio32/64 con $4Mb$. de memoria SVGA, un chip de BIOS de American Megatrends, una tarjeta de sonido SoundBlaster 16 y una tarjeta de red.
No presenta soporte para todos los programas, debido a que pueden existir fallos debido a errores en la sincronización de las operaciones o se pueden generar *opcode* fuera de tiempo.
La emulación en Macintosh es de recompilación dinámica para traducir código *x86* a código de un Mac con procesador PowerPC. En los Mac con procesador Intel no existe una versión de VirtualPC con lo que hay que acudir a otro tipo de soluciones.
La emulación en Windows también es de recompilación dinámica, pero solo traduce el modo de *kernel* y el modo real *x86* a código de usuario *x86*, mientras el usuario original corre en forma nativa o verdadera.

Ventajas:

- Familiar para usuarios Windows, integración con plataformas Microsoft correcta, soporte y documentación abundantes.
- Admiten drivers de los sistemas a emular.

Desventajas:

- Consumo excesivo de recursos, inestabilidad bajo ciertas condiciones de contorno, despliegue y ejecución lentos, virtualización dificultosa o imposible de algunos entornos derivados de UNIX.

- Su código es propietario y se comercializa bajo modelo de licencias.
- Únicamente dispone de versiones para Windows y Mac OS X.

2.2.5. Xen

Historia

Xen fue inicialmente un proyecto de investigación de la Universidad de Cambridge (la primer versión del software fue publicada a fines de 2003). Este proyecto de investigación fue liderado por Ian Pratt, quien luego formó una empresa -junto con otras personas- para dar servicios de valor agregado como soporte, mantenimiento y capacitación sobre Xen en Enero de 2005. Esta empresa es Xensource Inc., recibió fondos por millones de dolares de diferentes inversores y actualmente mantiene Xen (junto con otras empresas y la comunidad), también se dedica a programar aplicaciones adicionales no libres para facilitar el uso, instalación y mantenimiento de Xen.

Dado que Xen está licenciado bajo GPL el código no puede cerrarse, y no es solo Xensource quien mantiene el código, sino que varias empresas importantes como IBM, Sun, HP, Intel, AMD, RedHat, Novell están sumamente involucradas en el desarrollo asignando programadores al mantenimiento de este software.

Paravirtualización

En los ordenadores normalmente el software mas poderoso es el sistema operativo, ya que controla todos los recursos del CPU, como el uso compartido del mismo entre las aplicaciones, memoria virtual, entrada/salida a dispositivos, entre otras cosas. Estas tareas las puede realizar gracias a que los procesadores modernos soportan varios niveles de privilegios, cuatro exactamente. El sistema operativo, el supervisor, corre en el nivel 0 (más privilegiado) y las aplicaciones en nivel 3 (menos privilegiado).

¿Cómo se adapta Xen a este esquema?, utiliza una técnica llamada *ring deprivileging*, donde el sistema operativo es modificado para poder ejecutarse en nivel 1 dejando el nivel 0 para el Xen, el cual instala un pequeño módulo conocido como *hypervisor*. Este mecanismo le permite a Xen tener más poder que el sistema operativo controlando los recursos a los cuales este puede acceder. Este esquema de usos de niveles es lo que se llama 'paravirtualización', como se ha comentado al principio de la sección. Ver página 29.

El rendimiento superior de Xen es una de sus características principales, ya que los sistemas virtualizados corren directamente sobre el procesador, sin emulación. Sistemas de virtualización completa como VMware y Virtual PC o Virtual Server de Microsoft utilizan una técnica conocida como *binary translation*, donde las instrucciones privilegiadas

son reemplazadas con fragmentos de código que simulan las mismas, esta técnica es muy compleja y provoca grandes perdidas de rendimiento, sobre todo en aplicaciones con un uso intensivo de dispositivos.

Soporte de sistemas operativos

Actualmente se puede usar tanto como sistema anfitrión o invitado casi cualquier distribución de Linux, además existen *ports* para NetBSD y OpenSolaris. Con los sistemas propietarios ocurre lo contrario, debido a que se necesita modificar el núcleo para que se ejecute este con un nivel de privilegios distinto, así que ninguno de la familia Windows se puede usar como invitado.

Sin embargo, esta limitación es eliminada con las nuevas tecnologías de virtualización de Intel y AMD, que permiten ejecutar sistemas operativos en nivel del procesador 0 sin necesidad de modificación alguna, dejando un nivel de privilegio especial para el *hypervisor*. Este nivel de privilegio especial se llama *root-mode*, el resto de los componentes corren en *non-root-mode*.

Funciones del *hypervisor*

El término *hypervisor* viene de supervisor, que es como se llama al software que maneja las maquinas virtuales en la virtualización completa (VMware Workstation). Así este termino llevado al extremo sería el *hypervisor* que se está comentando. Se podría describir como un *microkernel* con las siguientes funciones:

- Planificación del tiempo de CPU.
- Protección de memoria entre máquinas virtuales.
- Encaminamiento de interrupciones.
- Mantenimiento del tiempo.
- Paso de mensajes entre máquinas virtuales.

De tal manera el *hypervisor* se ejecuta por debajo por debajo incluso del sistema operativo anfitrión proporcionando estabilidad, aislamiento entre maquinas y políticas de QoS (Calidad del servicio). La empresa VMware tiene en uno de sus productos un *hypervisor* con funciones parecidas a las de Xen (VMware Server ESX).

Paravirtualizacion VS *MicroKernels*

Es común pensar en la idea de desarrollar un *microKernel* en lugar de un *hypervisor* sobre el que corran las máquinas virtuales. Teóricamente no hay mucha diferencia, e incluso sería mejor ya que no se necesitaría tener corriendo ningún sistema operativo anfitrión consumiendo recursos del sistema.

La razón de esto es mas bien práctica, si se desarrolla un *microKernel* se deberá desarrollar también controladores para la máquina y esto supone un costo muy grande, casi imposible para cualquier proyecto, además de ser uno de los principales impedimentos para que triunfe cualquier nuevo sistema operativo. De aquí que se use un sistema operativo anfitrión como intermediario solventando este problema.

Funcionamiento de Xen

Arquitectura

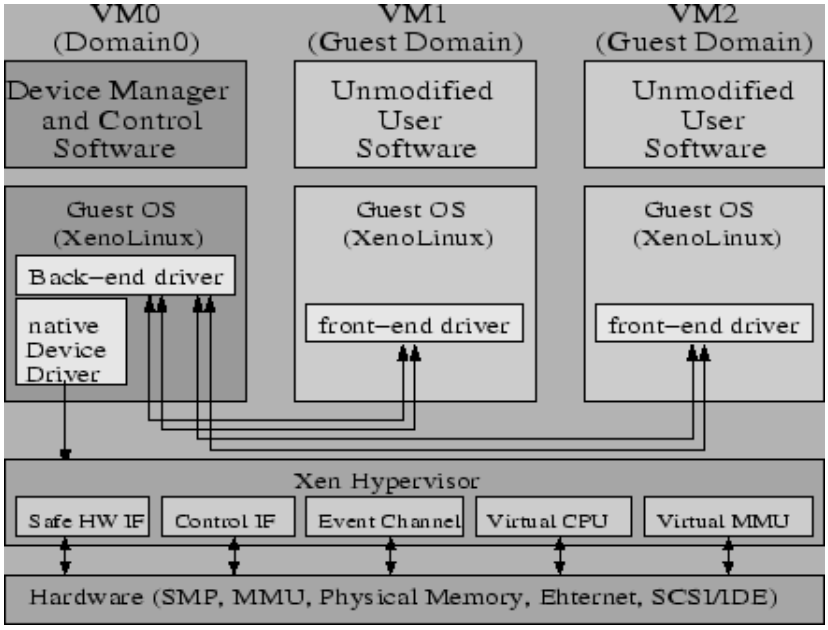


Figura 2.4: Arquitectura de xen

Es preciso aclarar la nomenclatura que utiliza Xen para los sistemas ejecutados: la maquina virtual anfitrión VM0 se suele notar como dominio 0 y las demás como dominio U.

En la máquina anfitrión el *kernel* se ejecuta en el nivel del microprocesador 0 así como el monitor o *hypervisor*. El resto de máquinas virtuales se ejecutan en el nivel 1. Las aplicaciones del todas las máquinas virtuales se ejecutan en nivel 3 el nivel con menores

privilegios.

En paravirtualización todas las máquinas virtuales usan el procesador directamente haciendo el *hypervisor* de planificador del tiempo de ejecución de tal manera que las maquinas corran de forma nativa.

Cuando las máquinas virtuales acceden a un dispositivo este acceso pasa por el dominio 0 proporcionando el aislamiento necesario. De esta forma los controladores de los dispositivos de las máquinas virtuales no son más que una API que se comunica con el anfitrión, este posee la implementación de los controladores de los dispositivos virtuales que no son más que unas traducciones hacia los controladores nativos, realiza la operación, accediendo al dispositivo y devuelve el resultado a la máquina invitada.

Uno de los puntos más conflictivos de que se ejecuten las instrucciones de las máquinas virtuales nativamente sobre el procesador son las interrupciones por falta de página. Para esto el *hypervisor* genera una CPU virtual y una unidad de gestión de memoria (MMU) virtual, pudiendo así correr las máquinas con más o con menos procesadores que los reales.

Las aplicaciones que hacen uso de muchas interrupciones hardware provocan una caída de rendimiento debido a la intercesión del *hypervisor* entre el hardware real y el virtual, así que Xen intenta minimizar al máximo estas actuaciones.

Como también se puede ver en la figura 2.4, Xen permite la ejecución de sistemas operativos sin modificación, hasta hace poco esto no era posible debido a la necesidad de que el *kernel* de estos se ejecutase a nivel de privilegio del procesador 1 en vez de 0, pero desde la salida de los últimos microprocesadores esto ya es posible por la inclusión de unas nuevas instrucciones al procesador, VT en el caso de Intel y Pacifica en el caso de AMD, que crean un nuevo nivel de privilegios por debajo del nivel 0 denominado *root-mode*, a partir de este nivel ya se pueden ejecutar cualquier sistema operativo sin modificación.

A continuación se describirá el modo de operación de Xen.

Interfaz de la máquina virtual

Los temas a tratar por el *hypervisor* son los siguientes:

- **Gestión de memoria**:
    - Segmentación: No se pueden usar descriptores de segmentos con todos los privilegios y tampoco se pueden superponer los segmentos con el final del espacio de direcciones.

  - Paginación: El sistema operativo invitado tiene acceso directo a las tablas de paginación (TLB) pero para actualizarla debe validarlo el *hypervisor*.

- **CPU**:
  - Protección: El sistema operativo invitado debe correr en un nivel de privilegios menor que el *hypervisor*.
  - Excepciones: El SO invitado debe registrar una tabla de manejadores de excepciones en Xen, de tal manera que, por ejemplo, las faltas de página las ejecute el *hypervisor*.
  - Llamadas al sistema: Las llamadas al sistema se ejecutan directamente, para esto previamente se deben validar, de tal manera que se mantenga el aislamiento entre máquinas virtuales.
  - Interrupciones: las interrupciones se remplazan por eventos del sistema.
  - Tiempo: Cada máquina virtual tiene una interfaz de tiempo, para mantener la diferencia entre el tiempo real y el tiempo virtual.
- **Dispositivos de entrada y salida**: para los dispositivos virtuales se capturan sus interrupciones hardware y se sustituyen por un mecanismo de eventos.

Gestión de memoria

Virtualizar la memoria es la parte mas difícil, requiere la intervención del *hypervisor* además de la modificación de cada sistema operativo invitado. El primer problema es la virtualización de la TLB, en otras arquitecturas se puede manejar por software, de tal manera que puedan coexistir diferentes TLBs de un modo eficiente, pero la arquitectura *x86* no lo permite, lo que conlleva que para cada modificación en esta deba de ser capturada y validada.

Sabiendo esto la solución a la que se ha llegado consta de dos puntos:

1. El sistema operativo invitado es responsable de manejar y alojar las tablas de paginación, con la intervención de Xen para asegurar el aislamiento y la seguridad.
2. Xen residirá en los últimos 64 MB del espacio de direcciones de cada máquina virtual para que la intervención en la TLB no conlleve un cambio de contexto hacia el *hypervisor*.

Cada vez que el sistema operativo invitado requiera alojar una nueva página en memoria esta se registrará en Xen, lo que quiere decir que el sistema invitado debe renunciar a escribir directamente en la tabla de paginación, lo que conlleva una modificación del sistema operativo.

La segmentación se gestiona de un modo similar, las únicas restricciones que se imponen a los descriptores de segmento del sistema operativo invitado son que: deben tener menor privilegio que Xen y que no se debe permitir ningún acceso a la porción de memoria reservada por este.

CPU

La virtualización de la CPU tiene importantes connotaciones, la primera es que Xen debe correr en un nivel de privilegios mayor que los sistemas operativos, lo que viola la suposición de que el sistema operativo debe ser la entidad con mayores privilegios en la máquina.

Así dado que la arquitectura x86 consta de cuatro niveles de privilegios (o rings), el *hypervisor* se situaría en el nivel 0, el de mayores privilegios, los sistemas operativos invitados en el nivel 1 y las aplicaciones que corran sobre estos en el nivel 3, el de menores privilegios. El uso de los tres niveles de privilegios permite garantizar la seguridad de que ni las aplicaciones podrán ejecutar instrucciones en el modo *kernel* del sistema operativo ni el sistema operativo podrá ejecutar las instrucciones privilegiadas del *hypervisor*, proporcionando un nivel de seguridad entre las distintas máquinas virtuales y el *hypervisor*. El problema para usar esta técnica en otras arquitecturas, es que algunas solo poseen dos niveles de privilegios, lo que provoca que el sistema operativo corra al mismo nivel que las aplicaciones, es decir, que se pierda el aislamiento entre el *kernel* del sistema operativo y las aplicaciones.

Las instrucciones que solo podría ejecutar Xen serían las relacionadas con las tablas de páginas y otras como *halt* que sirve para detener el procesador.

Las excepciones son tratadas de un modo bastante sencillo, una tabla contiene los punteros a las rutinas de cada excepción, esta tablas la registra Xen tras validarla. Esto es posible debido a que la gran mayoría de las rutinas son idénticas a las que se usarían directamente sin virtualización. Las rutinas que no son iguales son las que se han explicado antes, las relacionadas con memoria, para asegurar el aislamiento. Cuando se intenta ejecutar una instrucción fuera del nivel 0 la rutina de Xen crea una copia del marco de pila de esta en el sistema operativo invitado y le pasa el control a la excepción registrada por Xen.

Normalmente solo hay dos tipos de excepciones que puedan afectar notablemente el rendimiento del sistema por su frecuencia, las llamadas al sistema y las faltas de página. La solución que se utiliza en el caso de las llamadas al sistema es simplemente revisarlas para que se puedan ejecutar a nivel de privilegios 1 y dejar que se ejecuten directamente. Las faltas de página son un caso distinto ya que solo se pueden ejecutar en nivel 0 lo

que implica que siempre las deba procesar Xen. El proceso de ejecución de estas es el siguiente: Desde el sistema operativo anfitrión se mira el fallo de página, se mira que el segmento al que pertenezca la página este cargado y que la carga de esta página no afecte a los segmentos marcados como estáticos por Xen, si el segmento no está en memoria se sale de la subrutina con *iret*, con lo que en el sistema invitado se detectaría una doble falta, y lanzaría la interrupción correspondiente.

Dispositivos de entrada y salida

En la virtualización completa se emulan completamente los comportamientos de los dispositivos de la máquina virtual, en la paravirtualización únicamente se crea una capa de abstracción sobre los dispositivos reales. Así Xen provee una interfaz de dispositivos genéricos con los que se interactúa. Cuando una máquina virtual utiliza un dispositivo la orden val al controlador de esta máquina virtual que no es más que una interfaz del controlador real que está en el sistema operativo anfitrión, aquí se traduce la petición al los drivers nativos de los dispositivos físicos y se ejecuta la orden.

Esto aunque parezca que es lo mismo que en otras plataformas de virtualización completa como VMware Workstation, no es así, por ejemplo, en el caso del disco duro puede ser una partición real o por LVM, en la virtualización completa el disco duro no es más que un archivo de nuestro sistema de ficheros.

Otro ejemplo de esto sería la tarjeta gráfica, mientras que en la virtualización completa es impensable ejecutar juegos en 3D, la tarjeta gráfica virtual de Xen es una S3 Savage con soporte completo OpenGL que se ejecuta a la velocidad de la tarjeta real, habiéndose hecho pruebas de rendimiento con una perdida menor del 10

Interfaces de red virtuales en un sistema Xen

Xen crea, por defecto, siete pares de interfaces *ethernet* virtuales interconectadas para que *dom0* las utilice. Se pueden concebir como dos interfaces ethernet conectados por un cable *ethernet* cruzado interno. *Veth0* está conectada a *vif0.0*, *veth1* está conectada a *vif0.1*, y así sucesivamente hasta *veth7*, que está conectada a *vif0.7*. Pueden accederse o usarse configurando una dirección IP y una MAC en el costado de la *veth#* y luego enlazando el extremo *vif0.#* al puente. La figura 2.5, muestra esta configuración.

Cada vez que se crea una instancia *domU*, ésta recibe un identificador numérico (asignado automáticamente y sin la posibilidad de que el usuario lo elija). El primer *domU* será el número 1, el segundo el número 2, incluso aunque el número 1 ya no se esté ejecutando, etc.

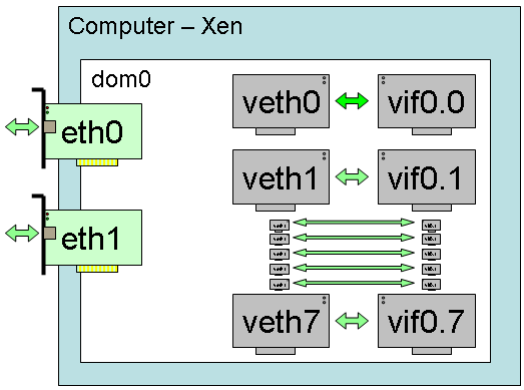


Figura 2.5: Interfaces de red virtuales en Xen

Para cada nuevo *domU*, Xen crea un nuevo par de interfaces ethernet virtuales conectados, con un extremo de cada par dentro del *domU* y el otro en el *dom0*. Si el *domU* usa Linux, el nombre de dispositivo se mostrará como *eht0*. El otro extremo de ese par de interfaces ethernet virtuales aparecerá dentro del *dom0* como interfaz *vif#.0*. Por ejemplo, la interfaz *eth0* del domU número 5 está conectada a vif5.0. Si se crean múltiples interfaces de red dentro de un domU, sus extremos se verán como eth0, *eth1*, etc, mientras que dentro de *dom0* aparecerán como *vif#.0*, *vif#.1*, etc. La figura 2.6, muestra las tarjetas de red lógicas conectadas entre el *dom0* y *dom1*.

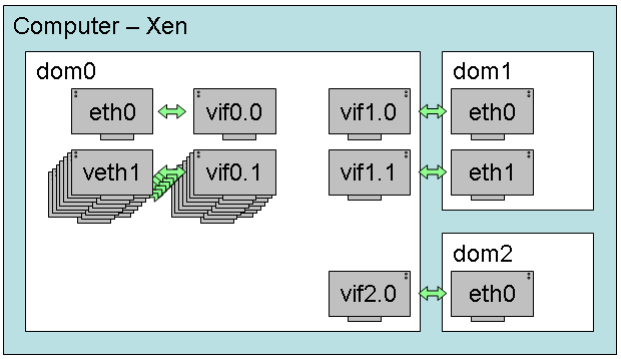


Figura 2.6: Tarjetas de red virtuales entre dominios Xen

Cuando un *domU* se detiene (usando el comando *xm shutdown domId*, por ejemplo), los interfaces *ethernet* virtuales que se crearon son eliminados.

Puentes de red en un sistema Xen

La configuración por defecto de Xen crea puentes de red (del inglés, *bridges* o *bridging*) dentro de *dom0* para permitir que todos los dominios aparezcan en la red como *hosts*

independientes.

Cuando un paquete llega al hardware, el controlador *ehternet* del dominio 0 lo gestiona y aparece en la interfaz *peth0*. *Peth0* está ligado al puente, por lo que es transferido al puente desde ahí. Este paso se ejecuta a nivel *ethernet* (no hay ninguna dirección IP establecida en la *peth0* o en el puente).

Acto seguido el puente distribuye el paquete del mismo modo que lo haría un concentrador (del inglés, *switch*). Luego, de entre las interfaces *vifX.Y* conectadas al puente se decide a dónde mandar el paquete basándose en la dirección MAC del receptor.

La interfaz *vif* pasa el paquete a Xen, el cuál a continuación lo envía de vuelta al dominio al cuál la *vif* apunta (también se hace así para el *dom0*, pues *vif0.0* está conectada a *veth0*). Finalmente, el dispositivo destinatario del *dom0/domU* tiene una dirección IP, por lo que se puede aplicar *iptables* aquí.

El script *network-bridge*

Cuando Xen arranca, ejecuta el script */etc/xen/scripts/network-bridge*, el cuál lleva a cabo las siguientes tareas:

- Crea un nuevo puente llamado *xenbr0*.
- Desactiva la interfaz *ethernet* real *eht0*.
- Copia las direcciones MAC e IP de la *eth0* a la interfaz virtual de red *veth0*.
- Renombra la interfaz real *eth0* a *peth0*.
- Renombra la interfaz virtual *veth0* a *eth0*.
- Conecta *peth0* y *vif0.0* al puente *xenbr0*.
- Activa el puente, *peth0*, *eth0* y *vif0.0*.

Es conveniente tener la interfaz física y la interfaz del *dom0* separadas, pues así es posible crear un cortafuegos en el *dom0* que no afecte al tráfico de los dominios *domU* (que proteja únicamente el *dom0*).

El *script vif-bridge*

Cuando arranca un *domU*, *xend*, que se está ejecutando en *dom0*, lanza el script *vif-bridge*, el cuál lleva a cabo las siguientes tareas:

- Enlaza la interfaz *vif#.0* al puente *xenbr0*.
- Levanta la interfaz *vif#.0*.

Xen 3

La última versión lanzada de Xen es la 3, que incluye las siguientes mejoras:

- **Soporte para máquinas SMP virtuales**: aunque Xen ha tenido soporte para múltiples procesadores desde hace tiempo, los dominios estaban restringidos a una única CPU. La versión 3 modifica esto. Se puede incluso cambiar el número de CPUs virtuales en tiempo de ejecución.
- **Soporte ACPI**: en Xen 2, el *hypervisor* tan sólo disponía de un soporte rudimentario de ACPI (aunque esto era suficiente para manejar la mayoría de las tablas de rutinas de las IRQ). En Xen 3 el dominio 0 tiene acceso a la mayoría de las funciones ACPI.
- **Soporte de hardware mejorado**: algunos problemas específicamente con el soporte AGP y DRM (gráficos 3D) de Linux fueron eliminados de las versiones anteriores 2.0.x y 3.0 *developer*.
- **Soporte PAE36**: Xen 2 estaba restringido al espacio de direcciones normal de 32 bits (4GB) en máquinas x86. La extensión PAE36 permite ahora que Xen pueda acceder a 64GB; desde luego, suponiendo que tanto Xen como el *kernel* han sido compilados con el soporte PAE incluido.
- **Soporte x86/64**: la variante de 64 bits de la arquitectura *x86* elimina todas las restricciones asociadas con el espacio de direcciones de 32 bits y añade diversas optimizaciones. Xen 3 soporta ahora sistemas operativos de 64bits.
- **Múltiples arquitecturas**: el soporte para IA64 (Itanium) está integrado en Xen 3, con soporte para la arquitectura Power.
- **Uso de las nuevas tecnologías de CPU**: Intel dispone de una nueva generación de CPUs con la extensión especial conocida como Tecnología Vanderpool. Por otro lado AMD dispone de la tecnología AMD-V en sus últimos procesadores. Estas extensiones facilitan y aceleran la implementación de una virtualización completa.

Conclusiones

Xen es un producto probado, utilizado y listo para usar en producción. El hecho de que esté licenciado bajo GPL no solo baja mucho los costos, sino que también le da mucha flexibilidad y proyección a futuro (en la vida del proyecto). Este producto combinado

con sistemas de almacenamiento propietarios o libres (utilizando LVM -Logical Volumen Manager, Mdadm -manejo de RAID por software para Linux-, IET -iSCSI Enterprise Target- y GFS -Global FileSystem-) puede dar excelentes resultados, alta disponibilidad y una escalabilidad sorprendente. Las ventajas son las siguientes:

- Independencia entre los sistemas virtualizados. Se pueden reiniciar, borrar y crear independientemente.
- Mejor aprovechamiento del hardware de la maquina: balanceo de recursos. Un sistema virtual puede recibir más recursos si los necesita y los demás sistemas no los necesitan.
- Facilidad para realizar copias de seguridad, solo es necesario copiar la máquina virtual permitiendo posteriormente que sea arrancada en un nuevo servidor. Xen incluso permite la migración en caliente, dando flexibilidad y escaso o nulo tiempo de recuperación ante un incidente.
- Se pueden modificar parámetros como la memoria RAM, el número de CPUs, espacio en disco... para ajustarlos a las necesidades de la máquina virtual.
- Se pueden crear máquinas de pruebas similares a las definitivas sin necesidad de adquirir hardware adicional.

Por otro lado tiene las siguientes desventajas:

- No soporta *drivers* propietarios de algunas tarjetas gráficas en el *dom0*, aunque se han publicado algunos parches para tener soporte a las distintas versiones del *driver* propietario de nVidia.
- Las versiones de pago son aquellas que implementan interfaces de usuario más intuitivos.

2.3. Sistemas de autenticación

Un sistema multiusuario necesita un método de autenticación para sus usuarios. Cada persona conocerá su nombre de usuario y contraseña que introducirá cuando necesite acceso al sistema. La base de datos que conforma toda la información de autenticación de los usuarios y el método utilizado para que se validen forman el sistema de autenticación. A continuación se realizará un breve estudio de los sistemas más utilizados.

2.3.1. NIS

NIS, acrónimo en inglés de Network Information Service que significa Sistema de Información de Red es un protocolo de servicios de directorios cliente-servidor desarrollado por Sun Microsystems para los datos de configuración de sistemas distribuidos tal como nombres de usuarios y *hosts* entre computadoras sobre una red. NIS está basado en *Sun RPC*, y consta de un servidor, una biblioteca de la parte cliente, y varias herramientas de administración.

Originalmente NIS se llamaba Páginas Amarillas (Yellow Pages), o YP, que todavía se utiliza para referirse a él. Desafortunadamente, ese nombre es una marca registrada de British Telecom, que exigió a Sun abandonar ese nombre. Sin embargo YP permanece como prefijo en los nombres de la mayoría de las órdenes relacionadas con NIS, como *ypserv* e *ypbind*. NIS proporciona prestaciones de acceso a bases de datos genéricas que pueden utilizarse para distribuir, por ejemplo, la información contenida en los ficheros */etc/passwd* y */etc/group* a todos los nodos de su red. Esto hace que la red parezca un sistema individual, con las mismas cuentas en todos los nodos. De manera similar, se puede usar NIS para distribuir la información de nombres de nodo contenida en */etc/hosts* a todas las máquinas de la red.

NIS esta implementado en prácticamente todas las distribuciones UNIX, e incluso existen implementaciones libres, que son las utilizadas en Linux.

Ventajas:

- Utiliza un sistema de directorios, obteniendo un buen rendimiento en las búsquedas, que son la mayoría de las operaciones realizadas.

Desventajas:

- Existen algunos productos que dan soporte NIS sobre windows, pero son proyectos de pago o con soporte muy limitado. Una solución es utilizar SAMBA como intermediario.
- No utiliza encriptación sobre los datos que envía por la red.

2.3.2. MySQL (enfocado al la autenticación)

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Aunque esta pensado para utilizarse con bases de datos relacionales, también es posible usarlo como base de datos de autenticación para un sistema UNIX. Las librerías PAM de dichos sistemas contienen un módulo para MySQL, de tal forma que se pueden hacer consultas (*querys*) a una base de datos que contendrá los nombres de usuario y contraseñas

del sistema.

Los usuarios son virtuales, mediante el modulo PAM-MySQL se les permite el acceso al sistema con o sin *shell*.
Ventajas:

- Al ser usuarios son virtuales, se pueden utilizar únicamente para un servicio de correo, sin tener acceso al sistema.

Desventajas:

- Rendimiento menos óptimo que un sistema de directorio, MySQL esta pensado para bases de datos relacionales.
- No hay productos conocidos para utilizar MySQL como método de autenticación en Windows.Una solución es utilizar Samba como intermediario.

2.3.3. Ldap

Ldap (*Lightweight Directory Access Protocol*) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. Ldap también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) al que pueden realizarse consultas.

Un directorio es una base de datos, pero en general contiene información más descriptiva y más basada en atributos. Proporcionan una respuesta rápida a operaciones de búsqueda o consulta, además pueden tener capacidad de replicar información de forma amplia, con el fin de aumentar la disponibilidad y fiabilidad, y a la vez reducir tiempo de respuesta.

Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, certificados, etc).

En conclusión, Ldap es un protocolo de acceso unificado a un conjunto de información sobre una red.

Implementaciones:

- **Active Directory**: es el nombre utilizado por Microsoft (desde Windows 2000) como almacén centralizado de información de uno de sus dominios de administración. Bajo este nombre se encuentra realmente un esquema (definición de los campos que pueden ser consultados) Ldap versión 3 lo que permite integrar otros sistemas que soporten el protocolo. En este Ldap se almacena información de usuarios, recursos de la red, políticas de seguridad, configuración, asignación de permisos, etc.

- **OpenLdap**: se trata de una implementación libre del protocolo que soporta múltiples esquemas por lo que puede utilizarse para conectarse a cualquier otro Ldap. Tiene su propia licencia, la OpenLdap Public License. Al ser un protocolo independiente de la plataforma, varias distribuciones Linux y BSD lo incluyen, al igual que AIX, HP-UX, Mac OS X, Solaris, Windows (2000/XP) y z/OS.

Ventajas:

- Al estar basado en un sistema de directorios, es muy rápido en la lectura de registros.
- Permite replicar el servidor de forma muy sencilla y económica.
- Muchas aplicaciones de todo tipo tienen interfaces de conexión a Ldap y se pueden integrar fácilmente.
- Dispone de un modelo de nombres globales que asegura que todas las entradas son únicas.
- Usa un sistema jerárquico de almacenamiento de información.
- Permite múltiples directorios independientes.
- Funciona sobre TCP/IP y SSL.
- La mayoría de aplicaciones disponen de soporte para Ldap.
- La mayoría de servidores Ldap son fáciles de instalar, mantener y optimizar.

Desventajas:

- Protocolo de manejo de datos poco intuitivo, pero existen múltiples herramientas que facilitan su uso.

Introducción a la estructura de árbol

Tradicionalmente se han usado las estructuras de árbol para jerarquizar la información contenida en un medio. El ejemplo más claro es la estructura de carpetas (directorios) de un sistema operativo. Esta organización nos permite ordenar la información en subdirectorios que contienen información muy específica.

Otro ejemplo muy común son los servidores DNS que nos permiten acceder a distintos servicios concretos que representan un dominio, por ejemplo:

www.empresa.com, servidor www principal de la empresa
www.admin.empresa.com, servidor de administración

mail.empresa.com, servidor de correo de la empresa
us.mail.empresa.com, servidor secundario de correo en USA
es.mail.empresa.com, servidor secundario de correo en España

Definición de términos

Entradas

El modelo de información de Ldap está basado en entradas. Una entrada es una colección de atributos que tienen un único y global Nombre Distintivo (DN). El DN se utiliza para referirse a una entrada sin ambigüedades. Cada atributo de una entrada posee un tipo y uno o más valores. Los tipos son normalmente palabras nemotécnicas, como *cn* para *common name*, o *mail* para una dirección de correo.
La sintaxis de los atributos depende del tipo de atributo. Por ejemplo, un atributo *cn* puede contener el valor 'Jose Manuel Suarez'. Un atributo *email* puede contener un valor *jmsuarez@ejemplo.com*. El atributo *jpegPhoto* ha de contener una fotografía en formato JPEG.

Objetos

En Ldap, una clase de objetos define la colección de atributos que pueden usarse para definir una entrada. El estándar Ldap proporciona estos tipos básicos para las clases de objetos:

1. Grupos en el directorio, entre ellos listas no ordenadas de objetos individuales o de grupos de objetos.
2. Emplazamientos, como por ejemplo el nombre del país y su descripción.
3. Organizaciones que están en el directorio.
4. Personas que están en el directorio.

Una entrada determinada puede pertenecer a más de una clase de objetos. Por ejemplo, la entrada para personas se define mediante la clase de objetos *person*, pero también puede definirse mediante atributos en las clases de objetos *inetOrgPerson*, *groupOfNames* y *organization*. La estructura de clases de objetos del servidor determina la lista total de atributos requeridos y permitidos para una entrada concreta.

Atributos

Los datos del directorio se representan mediante pares de atributo y su valor. Por ejemplo el atributo *commonName*, o *cn* (nombre de pila), se usa para almacenar el nombre de una persona. Puede representarse en el directorio a una persona llamada José Suarez mediante:

- `cn: José Suarez`

Cada persona que se introduzca en el directorio se define mediante la colección de atributos que hay en la clase de objetos *person*.
Otros atributos:

- `givenname: José`
- `surname: Suarez`
- `mail: jmsuarez@ejemplo.com`

Los atributos **requeridos** son aquellos que deben estar presentes en las entradas que utilicen esa clase de objetos. Los atributos **permitidos** son aquellos que pueden estar de manera opcional en las entradas que utilicen esa clase de objetos.

Por ejemplo, en la clase de objetos *person*, se requieren los atributos *cn* y *sn*. Los atributos *description* (descripción), *telephoneNumber* (número de teléfono), *seealso* (véase también), y *userpassword* (contraseña del usuario) se permiten pero no son obligatorios.

Cada atributo tiene la definición de sintaxis que le corresponde. La definición de sintaxis describe el tipo de información que proporciona ese atributo:

1. *bin* binario.
2. *ces* cadena con mayúsculas y minúsculas exactas (las mayúsculas y minúsculas son significativas durante las comparaciones).
3. *cis* cadena con mayúsculas y minúsculas ignoradas (las mayúsculas y minúsculas no son significativas durante las comparaciones).
4. *tel* cadena de número de teléfono (como cis, pero durante las comparaciones se ignoran los espacios en blanco y los guiones'-').
5. *dn* '*distinguished name*' (nombre distintivo).

Tipos de Atributos

Una definición de tipo de atributo especifica la sintaxis de un atributo y cómo se ordenan y comparan los atributos de ese tipo. A continuación se muestran los campos que definen cada atributo:

- *OID*: identificador del objeto único.
- *NAME*: nombre del atributo.
- *DESC*: descripción del atributo.
- *OBSOLETE*: '*true*' si es obsoleto, '*false*' o ausente si no lo es.
- *SUP*: nombre del atributo padre del cual deriva, en el caso de que sea derivado.
- *EQUALITY*: define el método a utilizar en las comparaciones, como sensible a mayúsculas, a espacios, etc.
- *ORDERING*: define el método a utilizar en las ordenaciones. Mismo caso que el anterior.
- *SUBSTRING*: define el método a utilizar en las comparaciones con fragmentos del campo. Mismo caso que el anterior.
- *SYNTAX OID*: se utiliza un número OID para definir el tipo de dato de entrada (UNICODE, ISO..) y el espacio en caracteres máximo.
- *SINGLE-VALUE*: define si es un atributo multivaluado (*true*) o no (*false*).
- *COLLECTIVE*: define si es un atributo que puede aparecer varias veces (*true*) o no (*false*).
- *NO-USER-MODIFICATION*: *true* si el atributo es modificable por el usuario, *false* si no lo es.
- *USAGE*: descripción del uso del atributo.

Los tipos de atributos en el directorio forman un árbol de clases. Por ejemplo, el tipo de atributo *commonName* es una subclase del tipo de atributo *name*. Los conjunto de clases de objetos se agrupan formando *schemas*, que son ficheros utilizados por el demonio de Ldap para construir su árbol de clases. Un ejemplo de *schema* es el fichero llamado *inetorgperson.schema* que agrupa las clases de objetos que definen la información de las personas de una organización. Otro ejemplo de *schema* es el fichero llamado *samba.schema*, que agrupa las clases de objetos que definen la información utilizada por Samba para la autenticación mediante Ldap.

Ldif

Para importar y exportar información de directorio entre servidores de directorios basados en Ldap, o para describir una serie de cambios que han de aplicarse al directorio, se

usa en general el fichero de formato conocido como Ldif (formato de intercambio de Ldap).

Un fichero Ldif almacena información en jerarquías de entradas con sus correspondientes atributos, normalmente es un fichero ASCII. Un ejemplo de fichero Ldif:

```
dn: uid=jmsuarez,ou=People,dc=empresa,dc=com
uid: jmsuarez
cn: Jose Manuel Suarez
objectclass: account
objectclass: posixAccount
objectclass: top
loginshell: /bin/bash
uidnumber: 512
gidnumber: 300
homedirectory: /home/jmsuarez
gecos: Jose Manuel Suarez,,,,
userpassword: {crypt}LPnaOoUYN57Netaac
```

Como se puede notar, cada entrada está identificada por un nombre distintivo: DN *distinguished name* esta compuesto por el nombre de la entrada en cuestión, más la ruta de nombres que permiten rastrear la entrada hacia atrás hasta la parte superior de la jerarquía del directorio.

Integración de Ldap con otros sistemas

Ldap puede utilizarse como repositorio de datos para multitud de aplicaciones que disponen de soporte. A continuación se listan algunas de ellas:

- Radius.
- Samba.
- DNS.
- *Mail Transfer Agents* (servidores de correo).
- Libretas de direcciones (por ejemplo en Mozilla Thunderbird).
- Servidores FTP.
- Servidores de certificados de seguridad.

2.4. Resto de servicios

Otros servicios que se necesitarán en la solución propuesta vienen listados a continuación:

2.4.1. Web

Un servidor web es un programa que implementa el protocolo HTTP (*hypertext transfer protocol*). Este protocolo está diseñado para lo que llamamos hipertextos, páginas web o páginas HTML (*hypertext markup language*): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Un servidor web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador.
Existen en el mercado dos soluciones importantes de servidores web: Apache e IIS.

Apache

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor 'parcheado').
El servidor Apache se desarrolla dentro del proyecto HTTP Server (*httpd*) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70 % de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

IIS

Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a

ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web).

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

2.4.2. Correo

Un servidor de correo es una aplicación que nos permite enviar mensajes (correos) de unos usuarios a otros, con independencia de la red que dichos usuarios estén utilizando. Para lograrlo se definen una serie de protocolos, cada uno con una finalidad concreta:

- **SMTP**, *Simple Mail Transfer Protocol*: Es el protocolo que se utiliza para que dos servidores de correo intercambien mensajes. Algunos de los más conocidos son Sendmail, Postfix, QMail, Exim y Microsoft Exchange Server.
- **POP**, *Post Office Protocol*: Se utiliza para obtener los mensajes guardados en el servidor y pasárselos al usuario.
- **IMAP**, *Internet Message Access Protocol*: Su finalidad es la misma que la de POP, pero el funcionamiento y las funcionalidades que ofrecen son diferentes.

Así pues, un servidor de correo consta en realidad de dos servidores: un servidor SMTP que será el encargado de enviar y recibir mensajes, y un servidor POP/IMAP que será el que permita a los usuarios obtener sus mensajes.
Para obtener los mensajes del servidor, los usuarios se sirven de clientes, es decir, programas que implementan un protocolo POP/IMAP. En algunas ocasiones el cliente se ejecuta en la máquina del usuario (como el caso de Mozilla Thunderbird, Evolution, Microsoft Outlook). Sin embargo existe otra posibilidad: que el cliente de correo no se ejecute en la máquina del usuario; es el caso de los clientes vía web, como GMail, Hotmail, OpenWebmail, SquirrelMail o Terra.

2.4.3. DNS

Domain Name System (DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como

base de datos el DNS es capaz de asociar distintos tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

Inicialmente, el DNS nació de la necesidad de recordar fácilmente los nombres de todos los servidores conectados a Internet. En un inicio, SRI (ahora SRI International) alojaba un archivo llamado HOSTS que contenía todos los nombres de dominio conocidos (técnicamente, este archivo aún existe - la mayoría de los sistemas operativos actuales todavía pueden ser configurados para revisar su archivo *hosts*).

El crecimiento explosivo de la red causó que el sistema de nombres centralizado en el archivo HOSTS no resultara práctico y en 1983, Paul Mockapetris publicó los RFCs 882 y 883 definiendo lo que hoy en día ha evolucionado al DNS moderno.

2.4.4. DHCP

DHCP (*Dynamic Host Configuration Protocol*) es un protocolo de red que permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme estas van estando libres, sabiendo en todo momento quien ha estado en posesión de esa IP, cuanto tiempo la ha tenido, a quien se la ha asignado después. También es posible asignar siempre las mismas IPs a las mismas máquina, si se mantiene una lista con las direcciones MAC de dichas máquinas y la IP asignada a cada una de ellas.

Existen implementaciones de DHCP en dispositivos hardware de red como routers, o firewalls. Por otro lado existen implementaciones propietarias como el servidor DHCP incorporado en los servidores de Microsoft Windows e implementaciones libres como en el caso de Linux.

2.4.5. Bases de datos

Las bases de datos son un servicio frecuentemente utilizado en ámbitos de servidores. Dan solución a la necesidad de almacenar datos siguiendo una determinada lógica y automatizando el tratado de los mismo. Se pueden clasificar de acuerdo a su modelo de administración de datos:

- **Bases de datos jerárquicas**: éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado

raíz, y a los nodos que no tienen hijos se los conoce como hojas. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. LDAP es un ejemplo de base de datos de este tipo.

- **Base de datos de red**: éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

- **Base de datos relacional**: éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Edgar Frank Codd postulo las bases del modelo relacional, su idea fundamental es el uso de "relaciones", estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Una manera más sencilla de conceptualizar una base de datos relacional a pesar de la teoría formulada, es entender cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).
  En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información. El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.
  Los sistemas de gestión de base de datos con soporte SQL más utilizados son:

  - **DB2**: propiedad de IBM.
  - **Oracle**: propiedad de Oracle Corporation, se considera uno de los sistemas de bases de datos más completos, pero su mayor defecto es su elevado precio, dado que es software propietario.
  - **SQL Server**: propiedad de Microsoft, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.
  - **Sybase ASE**: propiedad de Sybase, es multiplataforma, existiendo una edición gratuita para Linux, pero con límites de escalabilidad y almacenamiento.
  - **MySQL**: se ofrece bajo licencia GNU GPL, es multiplataforma. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

  - **PostgreSQL**: es un motor de base de datos, es servidor de base de datos relacional libre, liberado bajo la licencia BSD. Sus principales características son la alta concurrencia y la amplia variedad de tipos nativos.
  - **Firebird**: propiedad de la fundación Mozilla, es un sistema multiplataforma de administración de base de datos relacional de código abierto, basado en la versión 6 de Interbase.
  - **Informix**: se trata de una familia de productos de administración de bases de datos relaciones adquirida por IBM.

- **Bases de datos orientadas a objetos**: este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los objetos completos (estado y comportamiento). En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos.

- **Bases de datos documentales**: permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Tesaurus es un sistema de índices optimizado para este tipo de bases de datos.

- **Base de datos deductivas**: un sistema de base de datos deductivas, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática.

- **Gestión de bases de datos distribuida**: la base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada centro de procesamiento de datos.

2.4.6. Terminal remoto

Un terminal remoto se puede entender como una computadora que hace la función de servir como *front-end* para otra computadora que se encarga de realizar las operaciones o bien como una herramienta software que permite el acceso a una computadora que está fuera de nuestro alcance físico. Se tratará el segundo caso.

Hay dos tipos de herramientas cliente: las herramientas gráficas, que permiten visualizar e interactuar con un escritorio gráfico ejecutado en la máquina remota, y las herramientas por línea de comandos, que permiten ejecutar comandos en la máquina remota. En ambos casos ha de existir otra herramienta software que funciona a modo de servidor.

Herramientas gráficas:

- **Escritorio remoto de Microsoft** (*remote desktop*): se trata de un software de Microsoft que permite utilizar el escritorio de una máquina remota mediante la red. Para ello, la máquina remota ha de tener licencia de Terminal Server (en el caso de la familia Microsoft Windows 2000) o bien ser de la familia Microsoft Windows XP.

- **Rdesktop**: se trata de un cliente de código abierto que permite conectarse por red a cualquier máquina que ejecute el protocolo RPD (*Remote Desktop Protocol*).

- **VNC**: VNC es un programa de software libre basado en una estructura cliente-servidor el cual nos permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente. También llamado software de escritorio remoto. VNC permite que el sistema operativo en cada computadora sea distinto: Es posible compartir la pantalla de una máquina de 'cualquier' sistema operativo conectando desde cualquier otro ordenador o dispositivo que disponga de un cliente VNC portado.

Herramientas por línea de comandos:

- **RSH**, *Remote SHell*: herramienta basada en el protocolo rlogin que permite obtener una shell de un ordenador remoto. No utiliza ningún tipo de cifrado, lo que representa un riesgo de seguridad si se intercepta la comunicación. No existe este comando para Windows.

- **SSH**, *Secure SHell*: herramienta que permite acceder a una máquina remota por red. Es capaz de obtener una *shell* de un ordenador remoto, de transferir datos entre éste y la máquina cliente y de redirigir tráfico de X para poder ejecutar programas gráficos. La seguridad es elevada dado que SSH utiliza técnicas de cifrado desde el inicio de la conexión.

2.4.7. Sistema de archivos en red

Los sistemas de ficheros distribuidos son necesarios en redes de varias computadoras en las cuales queremos tener unicidad en los datos de una manera transparente al usuario. El usuario necesita acceder a sus datos desde cualquier computador y ver un único sistema de ficheros, aunque por debajo, el sistema distribuya dichos datos entre varias computadoras. Una de las soluciones más antiguas y que sigue siendo notable es NFS (*Network File System*) que posibilita el acceso a ficheros remotos como si se tratase de locales. Originalmente fue desarrollado en 1984 por Sun Microsystems, con el objetivo de que sea independiente de la máquina, el sistema operativo y el protocolo de transporte,

esto fue posible gracias a que está implementado sobre los protocolos XDR (presentación) y SUN RPC (sesión). El protocolo NFS está incluido por defecto en los Sistemas Operativos UNIX y las distribuciones Linux.

Características:

- El sistema NFS está dividido al menos en dos partes principales: un servidor y uno o más clientes. Los clientes acceden de forma remota a los datos que se encuentran almacenados en el servidor.

- Las estaciones de trabajo locales utilizan menos espacio de disco debido a que los datos se encuentran centralizados en un único lugar pero pueden ser accedidos y modificados por varios usuarios, de tal forma que no es necesario replicar la información.

- Los usuarios no necesitan disponer de un directorio *home* en cada una de las máquinas de la organización. Los directorios *home* pueden crearse en el servidor de NFS para posteriormente poder acceder a ellos desde cualquier máquina a través de la infraestructura de red.

- También se pueden compartir a través de la red dispositivos de almacenamiento como disqueteras, CD-ROM y unidades ZIP. Esto puede reducir la inversión en dichos dispositivos y mejorar el aprovechamiento del hardware existente en la organización.

Todas las operaciones sobre ficheros son síncronas. Esto significa que la operación sólo retorna cuando el servidor ha completado todo el trabajo asociado para esa operación. En caso de una solicitud de escritura, el servidor escribirá físicamente los datos en el disco, y si es necesario, actualizará la estructura de directorios, antes de devolver una respuesta al cliente. Esto garantiza la integridad de los ficheros.

Versiones:

- La versión 2 de NFS (NFSv2), es la más antigua y está ampliamente soportada por muchos sistemas operativos.

- La versión 3 de NFS (NFSv3) tiene más características, incluyendo manejo de archivos de tamaño variable y mejores facilidades de informes de errores, pero no es completamente compatible con los clientes NFSv2.

- NFS versión 4 (NFSv4) incluye seguridad Kerberos, trabaja con cortafuegos, permite ACLs y utiliza operaciones con descripción del estado.

2.5. Aumento de fiabilidad de datos

Las soluciones de aumento de fiabilidad de datos permiten que los datos de un sistema sean siempre fiables, aun en presencia de incidencias de cualquier tipo. Para conseguir esta fiabilidad hay varias herramientas software y hardware que se describirán a continuación.

2.5.1. RAID

El acrónimo RAID (*Redundant Array of Independent Disks*, 'conjunto redundante de discos independientes') hace referencia a un sistema de almacenamiento informático que usa múltiples discos duros entre los que distribuye o replica los datos. Dependiendo de su configuración (a la que suele llamarse 'nivel'), los beneficios de un RAID respecto a un único disco son uno o varios de los siguientes: mayor integridad, mejor tolerancia a fallos, más rendimiento y más capacidad.

Las configuraciones o niveles más importantes son:

- **RAID 0**: distribuye los datos equitativamente entre dos o más discos sin información de paridad o redundancia, es decir, no ofrece tolerancia al fallo (si ocurriese alguno, la información de los discos se perdería y debería restaurarse desde una copia de seguridad). El RAID 0 se usa normalmente para incrementar el rendimiento, aunque también puede utilizarse como forma de crear un pequeño número de grandes discos virtuales a partir de un gran número de pequeños discos físicos.

- **RAID 1**: crea una copia exacta (o espejo) de un conjunto de datos en dos o más discos (array). Esto resulta útil cuando el rendimiento en lectura es más importante que la capacidad y también desde el punto de vista de la seguridad, pues un RAID 0 por ejemplo no es tolerante al fallo de uno de los discos, mientras que un RAID 1 sí, al disponer de la misma información en cada disco.
  Un conjunto RAID 1 es tan grande como el más pequeño de sus discos. Un RAID 1 clásico consiste en dos discos en espejo, lo que incrementa exponencialmente la fiabilidad respecto a un solo disco; es decir, la probabilidad de fallo del conjunto es igual al producto de las probabilidades de fallo de cada uno de los discos (pues para que el conjunto falle es necesario que lo hagan todos sus discos).
  Adicionalmente, dado que todos los datos están en dos o más discos, con hardware habitualmente independiente, el rendimiento de lectura se incrementa aproximadamente como múltiplo linear del número del copias; es decir, un RAID 1 puede estar leyendo simultáneamente dos datos diferentes en dos discos diferentes, por lo que su rendimiento se duplica.
  Como en el RAID 0, el tiempo medio de lectura se reduce, ya que los sectores a buscar pueden dividirse entre los discos, bajando el tiempo de búsqueda y subiendo

la tasa de transferencia, con el único límite de la velocidad soportada por la controladora RAID
Al escribir, el conjunto se comporta como un único disco, dado que los datos deben ser escritos en todos los discos del RAID 1. Por tanto, el rendimiento no mejora.
El RAID 1 es un sistema apropiado en entornos donde la disponibilidad es crítica 24 horas al día. Aparte de los discos en espejo que crean el array en RAID 1 podemos marcar discos adicionales como reserva. Éstos se pueden definir como *spare disk* si queremos que estén en funcionamiento o *standby spare disk*, si queremos que estén en modo de espera. En el momento que alguno de los discos del espejo sufra algún fallo, uno de los discos de reserva entra a formar parte del array de discos espejo (entra instantáneamente si es un *spare disk* o tarda unos instantes si tiene que arrancar al ser un disco *standby spare disk*), duplicándose la información en él.

- **RAID 2**: divide los datos a nivel de bits en lugar de a nivel de bloques y usa un código de *Hamming* para la corrección de errores. Los discos son sincronizados por la controladora para funcionar al unísono. Éste es el único nivel RAID original que actualmente no se usa. Permite tasas de trasferencias extremadamente altas.
  Teóricamente, un RAID 2 necesitaría 39 discos en un sistema informático moderno: 32 se usarían para almacenar los bits individuales que forman cada palabra y 7 se usarían para la corrección de errores.

- **RAID 3**: usa división a nivel de *bytes* con un disco de paridad dedicado. El RAID 3 se usa rara vez en la práctica. Uno de sus efectos secundarios es que normalmente no puede atender varias peticiones simultáneas, debido a que por definición cualquier simple bloque de datos se dividirá por todos los miembros del conjunto, residiendo la misma dirección dentro de cada uno de ellos. Así, cualquier operación de lectura o escritura exige activar todos los discos del conjunto.

- **RAID 4**: usa división a nivel de bloques con un disco de paridad dedicado. El RAID 4 es parecido al RAID 3 excepto porque divide a nivel de bloques en lugar de a nivel de *bytes*. Esto permite que cada miembro del conjunto funcione independientemente cuando se solicita un único bloque. Si la controladora de disco lo permite, un conjunto RAID 4 puede servir varias peticiones de lectura simultáneamente. En principio también sería posible servir varias peticiones de escritura simultáneamente, pero al estar toda la información de paridad en un solo disco, éste se convertiría en el cuello de botella del conjunto.

- **RAID 5**: usa división de datos a nivel de bloques distribuyendo la información de paridad entre todos los discos miembros del conjunto. El RAID 5 ha logrado popularidad gracias a su bajo coste de redundancia. Generalmente, el RAID 5 se implementa con soporte hardware para el cálculo de la paridad.
  Los bloques de paridad no se leen en las operaciones de lectura de datos, ya que

esto sería una sobrecarga innecesaria y disminuiría el rendimiento. Sin embargo, los bloques de paridad se leen cuando la lectura de un sector de datos provoca un error de control de redundancia cíclica (CRC). En este caso, el sector en la misma posición relativa dentro de cada uno de los bloques de datos restantes en la división y dentro del bloque de paridad en la división se utilizan para reconstruir el sector erróneo. El error CRC se oculta así al resto del sistema. De la misma forma, si falla un disco del conjunto, los bloques de paridad de los restantes discos son combinados matemáticamente con los bloques de datos de los restantes discos para reconstruir los datos del disco que ha fallado «al vuelo».
Se necesita un mínimo de tres unidades para implementar una solución RAID 5. Las implementaciones RAID 5 presentan un rendimiento malo cuando se someten a cargas de trabajo que incluyen muchas escrituras más pequeñas que el tamaño de una división (*stripe*). Esto se debe a que la paridad debe ser actualizada para cada escritura, lo que exige realizar secuencias de lectura, modificación y escritura tanto para el bloque de datos como para el de paridad.

- **RAID 0+1**: un RAID 0+1 es un espejo de divisiones. Primero se crean dos conjuntos RAID 0 (dividiendo los datos en discos) y luego, sobre los anteriores, se crea un conjunto RAID 1 (realizando un espejo de los anteriores). La ventaja de un RAID 0+1 es que cuando un disco duro falla, los datos perdidos pueden ser copiados del otro conjunto de nivel 0 para reconstruir el conjunto global. Sin embargo, añadir un disco duro adicional en una división, es obligatorio añadir otro al de la otra división para balancear el tamaño del conjunto.
  Además, el RAID 0+1 no es tan robusto como un RAID 10, no pudiendo tolerar dos fallos simultáneos de discos salvo que sean en la misma división. Es decir, cuando un disco falla, la otra división se convierte en un punto de fallo único. Además, cuando se sustituye el disco que falló, se necesita que todos los discos del conjunto participen en la reconstrucción de los datos.

- **RAID 1+0**: es parecido a un RAID 0+1 con la excepción de que los niveles RAID que lo forman se invierte: el RAID 10 es un división de espejos.
  En cada RAID 1 pueden fallar todos los discos salvo uno sin que se pierdan datos. Sin embargo, si los discos que han fallado no se reemplazan, el restante pasa a ser un punto único de fallo para todo el conjunto. Si ese disco falla entonces, se perderán todos los datos del conjunto completo. Como en el caso del RAID 0+1, si un disco que ha fallado no se reemplaza, entonces un solo error de medio irrecuperable que ocurra en el disco espejado resultaría en pérdida de datos.
  El RAID 10 es a menudo la mejor elección para bases de datos de altas prestaciones, debido a que la ausencia de cálculos de paridad proporciona mayor velocidad de escritura.

Implementaciones

Los sistemas operativos suelen tener implementaciones RAID por software que emplean la prestaciones del procesador y pueden competir con las del hardware RAID, debido a la potencia de los procesadores actuales. Las soluciones hardware gestionan el subsistema RAID independientemente del *host*, presentándole a este un solo disco.

2.5.2. *Backup*

Las copias de seguridad en un sistema informático tienen por objetivo el mantener cierta capacidad de recuperación de la información ante posibles pérdidas. Suelen ser utilizadas como la última línea de defensa contra pérdida de datos, y se convierten por lo tanto en el último recurso a utilizar. Las copias de seguridad se pueden hacer de distintos tipos:

- **Completa**: realiza una copia completa de los datos de origen en el la zona de destino especificada.
- **Incremental**: se guardan sólo con los ficheros que se hayan modificado desde la última copia de seguridad, ya sea este último incremental o completo. Recuperar y restaurar un sistema completamente a un cierto punto en el tiempo requiere localizar una copia de seguridad completa y todas las incrementales posteriores realizadas hasta el instante que se desea restaurar. Los inconvenientes son tener que tratar con grandes series de copias incrementales y contar con un gran espacio de almacenaje.
- **Diferencial**: igual que en incremental, se guardan sólo con los ficheros que se hayan modificado desde la última copia de seguridad, salvo que ésta ha de ser completa.

Una solución software muy importante para realizar *backups* es la herramienta *rsync* de Linux, que permite sincronizar archivos y directorios entre dos máquinas de una red o entre dos ubicaciones en una misma máquina, minimizando el volumen de datos transferidos.

El medio físico utilizado para guardar las copias de seguridad puede ser de diversos tipos: un disco duro, un CD o DVD, cintas magnéticas de *backup* o dispositivos de memoria no volátil como las memorias Flash.

Existen varias reglas fundamental para una estrategia de *backup*:

- Tener las copias de seguridad en una localización física distinta a donde residen los datos originales.
- Es tan importante la copia de seguridad como su estrategia de recuperación asociada.

- Es preferible tener un sistema automático de realización de copias de seguridad para evitar errores humanos.
- Toda información de datos almacenada y cuya importancia sea muy alta necesitaría debería ser copiada.

Capítulo 3

Análisis

3.1. Situación de ARCOS

El grupo (ARCOS) del Área de Arquitectura y Tecnología de Computadores, cuenta con personal docente e investigador asi como alumnos becarios realizando proyectos. Para realizar sus tareas, el grupo ARCOS dispone tanto de despachos propios para los profesores titulares y personal investigador, como un laboratorio para el resto del personal. En cada despacho, el profesor dispone de un pc para realizar sus tareas asi como acceso a las impresoras compartidas en red. El grupo cuenta con un laboratorio destinados a las tareas de investigación, pruebas, proyectos, etc. Cuenta con equipos informáticos con el hardware y software específico para el trabajo en cada una de las áreas en las que está especializado, conectados a la Intranet del grupo y con conexión directa a Internet.

Dentro del laboratorio se localizaban dos servidores con Linux (*Aguila* y *Cuervo*) junto al servidor *Codorniz* con Windows Server 2003, actuando de controlador de dominio para el dominio existente en ARCOS (denominado WIN).

Estos servidores permitían trabajar desde máquinas con Windows o Linux, con la consiguiente autenticación. Las máquinas con Windows estaban dentro del dominio WIN cuyo controlador primario de dominio era *Codorniz*. Las máquinas con Linux utilizaban a *Cuervo* como servidor de autenticación y de disco (mediante NFS), conteniendo ésta solo las cuentas de aquellos usuarios que las utilizaban. El servidor *Aguila* contenía el resto de las cuentas, es decir, las de proyectos, asignaturas, profesores y personal investigador, teniendo como único punto de acceso a dichos datos un terminal remoto (SSH) a la misma. *Aguila* también ejecutaba la mayoría de los servicios que utilizaba el grupo, existiendo un elevado riesgo de seguridad y de disponibilidad de los datos al tener acceso por ssh la mayoría del personal de ARCOS.

A efectos prácticos, este riesgo se traducía en constantes bloqueos de la máquina y reinicios de la misma para volver a su estado operativo, afectando al rendimiento del personal, que necesita de los servicios que proporcionaba *Aguila*.

3.2. ¿Como estaba anteriormente el sistema?

A continuación se describirá como estaba diseñado el antiguo sistema para cumplir sus objetivos. Se analizarán los servicios que ofrecía y qué medidas tomaba respecto a la fiabilidad de los datos, la seguridad, la rapidez y la disponibilidad.

3.2.1. Servicios

En esta sección se exponen varios de los servicios que ofrecía el antiguo sistema de servidores y cómo los ofrecía:

- **Autenticación de usuarios, tanto para windows como para linux**: los servidores que autenticaban a los usuarios eran *Aguila*, *Cuervo* y *Codorniz*. *Aguila* era el servidor responsable de la autenticación del personal docente e investigador y de las cuentas de las asignaturas del área. Estas cuentas disponían de correo, acceso SSH al propio servidor, un espacio de almacenamiento y un espacio web, tanto en modo normal como en modo seguro. *Cuervo* autenticaba a los usuarios que utilizaban los ordenadores con Linux del laboratorio y por último *Codorniz* autenticaba a los usuarios que utilizaban las cuentas de Windows.

- **Almacenamiento para los usuarios y otros datos de los que dispone el área**: el almacenamiento de los usuarios también se dividía entre los tres servidores dedicados a la autenticación. *Aguila* almacenaba la información de las cuentas de usuario de aquellos que tenían correo, además de todos los datos de los proyectos realizados y asignaturas impartidas por el grupo y las webs correspondientes.
  Cuervo contenía la información de las cuentas creadas en él para utilizar los ordenadores con Linux, que montaban dichas cuentas y autenticaban a través de *Cuervo* utilizando NIS.
  Por último, *Codorniz*, el servidor Windows, en un principio utilizaba perfiles móviles, permitiendo tener los datos de la cuenta de Windows centralizados, pero más tarde se optó por los perfiles locales. Esta solución ofrecía mayor velocidad de acceso a una determinada cuenta pero dispersaba los datos de cada usuario entre las máquinas que había utilizado.

- **Servicio WEB**: cada cuenta de usuario contaba con un espacio web normal y otro espacio web seguro (*https*). La autenticación para aquellas webs seguras que lo solicitaban, se hacia mediante un modulo de *Apache* que accedía al fichero */etc/passw*d de *Aguila*.

- **Servicio DNS**: para la resolución de nombres, se crearon dos subdominios: *linux.arcos.inf.uc3m.es* y *win.arcos.inf.uc3m.es*, el primero para la resolución de las máquinas con Linux instalado y el segundo para la resolución de las máquinas con Windows instalado. Las máquinas que contenían ambos sistemas operativos estaban presentes en ambos subdominios. La resolución de nombres de *linux.arcos.inf.uc3m.es* la realizaba *Aguila* y la resolución de *win.arcos.inf.uc3m.es* la realizaba *Codorniz*. Además, se necesitaba otro servicio DNS que permitiese la delegación a cualquiera de los subdominios, existiendo un tercer servicio DNS encargado de ésta tarea instalado en *Aguila*.

- **Servicio de correo**: el servidor de correo estaba instalado en *Aguila*, usando Postfix, y no utilizaba ninguna herramienta para la detección de correo basura o ficheros infectados. Además existían los servicios de IMAP y POP3, tanto en modo normal como en modo seguro (SSL). El formato que se utilizaba era *mailbox*, que consiste en dejar los correos de cada cuenta en un solo fichero.
  Como además se utilizaba Squirrelmail como servicio de correo web, éste permitía la existencia de varias carpetas en el servidor, pero siguiendo el mismo esquema que *mbox*, es decir, cada carpeta contenía un solo fichero con todos los correos de la misma.
  El principal problema de esta solución era la lentitud con que se accedía al correo cuando los ficheros tenían un tamaño considerable. Otro de los problemas es que se corrompiera la cabecera de algún correo dentro de un fichero imposibilitando la lectura de los correos siguientes, obligando a editar el fichero manualmente para solucionar el problema.

- **Servicio de terminal remoto para linux (SSH)**: solo se disponía de acceso por SSH al servidor *Aguila*, donde residían los datos de cada cuenta de usuario, la web segura (*https*) y normal y los ficheros *mbox* de correo. El problema principal es que al tratarse de un servidor, era un riesgo muy elevado el acceder por SSH al mismo, dado que cualquier usuario podía ejecutar programas de la máquina o propios, con el riesgo de comprometer el sistema operativo, los servicios que se están ejecutando y los datos que contiene la máquina del resto de cuentas.

- **Servicios no ofrecidos**: no existía un servicio de DHCP, obligando al administrador a buscar una IP libre cuando alguien necesitaba conectar temporalmente una máquina con *ethernet* para tener acceso a la red de ARCOS.

3.2.2. Directrices del sistema

A continuación se muestra un resumen de cómo el antiguo sistema ofrecía las directrices a seguir en la reestructuración que se va a llevar a cabo. Las directrices son: fiabilidad de los datos, seguridad, rapidez y disponibilidad.

- Fiabilidad de los datos.
  - **Uso de RAID como solución de aumento de fiabilidad de datos**: el servidor *Aguila* disponía de RAID1 para el almacenamiento de las cuentas, no obstante tanto *Codorniz* como *Cuervo*, no disponían de RAID.
- Seguridad.
  - **Seguridad de los datos de asignaturas, proyectos y datos personales**: la seguridad de las cuentas de *Aguila*, residía en los permisos del sistema Linux. Dado que hay acceso SSH al propio servidor, cualquier usuario puede ejecutar un *exploit* para tener permisos de *root* en la máquina y asi poder ver la información de cualquier cuenta. Esto conlleva dos problemas, el primero es la imposibilidad de denegar la ejecución de ciertos programas y el segundo no conocer la identidad de quién ejecuta cada programa, dado que no existía ninguna herramienta de monitorización de los comandos ejecutados.
    En el caso de *Cuervo*, las máquinas que montaban por NFS las cuentas del mismo, lo hacían con *root_squash*, impidiendo que si un usuario conseguia privilegios de *root*, se metiera en el directorio de una cuenta que no fuese la suya propia. No obstante, también podía loguearse en el propio servidor *Cuervo*, existiendo los mismos riesgos que en *Aguila*.
    En el caso de *Codorniz*, los datos de cada usuario estaban en la máquina cliente donde se loguease, dependiendo de los permisos de Windows. Solo los usuarios con permisos de administrador local o del dominio podían ver los datos que había en la carpeta '*Documents and settings*'.
  - **Separación de clientes/servidores**: según el antiguo esquema, este punto sólo se cumplía para el servidor *Codorniz*. No se dejaba iniciar sesión a ningún usuario, salvo al administrador, de tal forma que ningún usuario reducía el rendimiento de la misma. Sin embargo, en el caso de *Aguila* y de *Cuervo*, cualquier usuario con cuenta en el propio servidor podía acceder al mismo, con el riesgo de ejecutar cualquier programa que bloquease la máquina, dejando de ser operativa y anulando todos los servicios que ambos servidores ofrecían.
- Rapidez.
  - **Automatización en la administración de usuarios**: en el caso de *Aguila* y *Cuervo*, se utilizaban los ficheros */etc/passwd* y */etc/shadow* para la autenticación de usuarios. Se realizaban *scripts* para automatizar las tareas de administración, y módulos de autenticación a través de las librerias *pam* en Apache, para las webs seguras y el servicio de correo web (*squirrelmail*).
    En el caso de *Codorniz*, se utilizaba Active Directory y las herramientas que integra para realizar las operaciones de administración.

El problema que existía en el antiguo sistema para obtener rapidez en la administración era la falta de unicidad en las cuentas de usuario, llegando el caso de tener tres cuentas distintas para la misma persona, una en *Aguila*, otra en *Cuervo* y otra en *Codorniz*.

- Disponibilidad.

  - **Facilidad de recuperación de las maquinas, los servicios y los datos críticos**: en el caso de *Aguila*, ésta contaba con una máquina réplica, en la que se realizaba una copia del sistema y de los datos de las cuentas de usuario manualmente. Esta solución no ofrecía una recuperación óptima de los servicios y datos críticos de *Aguila*, dado que dependía del tiempo transcurrido entre la copia de seguridad manual y el fallo de *Aguila*. Por su lado, el propio servidor *Aguila*, disponía de RAID1, ofreciendo tolerancia a fallos sobre cualquier incidencia en uno de los discos. No ocurre lo mismo con los servicios, teniendo que utilizar la máquina réplica.
    En el caso de *Cuervo*, no disponía ni de RAID, ni de máquina réplica, no admitiendo ningún tipo de fallo de la misma. Por último, *Codorniz*, tampoco disponía de copias de seguridad ni una máquina que sirviese de segundo controlador de dominio.

  - **Alta disponibilidad**: solo *Aguila* tenía una máquina réplica para poder ofrecer alta disponibilidad. No obstante, la sincronización de ambas se hacia de manera manual, al igual que la puesta en marcha de la máquina réplica en caso de fallo de *Aguila*.

  - **Separación física de las máquinas críticas**: los tres servidores críticos del antiguo sistema, es decir, *Aguila*, *Cuervo* y *Codorniz*, eran máquinas físicas distintas, no virtuales, no incidiendo la caída de cualquiera de las máquinas físicas en las otras.

    En cuanto a su localización física, las máquinas estaban en el laboratorio, bajo unas condiciones de temperatura poco recomendables dado que el aire acondicionado deja de funcionar todos los días desde las 19 hasta las 9 horas del día siguiente. Además, cualquier persona podía accidentalmente pulsar el botón de apagado de cualquier máquina o tirar del cable de alimentación o de red.

3.3. ¿Qué se necesita?

Para la restructuración del sistema de servidores de ARCOS se han analizado las necesidades a cubrir, que se pueden dividir en servicios a proporcionar y orientación a seguir para dichas necesidades.

3.3.1. Servicios

Los servicios que se necesitan son los siguientes:

- **Autenticación de cuentas de usuario, tanto para windows como para linux**: se necesita una solución que permita la autenticación de usuarios en Windows y en Linux utilizando la misma base de datos, con objeto de alcanzar una unicidad para la información de los usuarios.

- **Almacenamiento para las cuentas de usuarios y otros datos de los que dispone el área**: se necesita un espacio de disco para cada cuenta de usuario que mantenga un nivel de seguridad y de privacidad óptimo, así como fiabilidad de los datos mediante *backups*.

- **Servicio WEB**: se necesita un espacio web para cada cuenta de usuario y un espacio web seguro (*https*) para la información privada de cada cuenta con acceso web (transparencias de asignaturas por ejemplo).

- **Servicio DNS**: se necesita un servicio DNS para la resolución de nombres de todas las máquinas, tanto Windows como Linux. Además, debe permitir la existencia de un servidor secundario.

- **Servicio de correo**: se necesita un servidor de correo que permita el envío y la recepción de mensajes dentro de ARCOS. Los protocolos a utilizar son POP3 e IMAP tanto en modo normal como modo seguro (SSL) y un servidor SMTP. También se necesita un cliente web para permitir el acceso mediante un navegador.
  Debe tener filtros *anti-spam* así como un antivirus instalado, para reducir al mínimo la intrusión de software maligno. Se necesita además que la velocidad del correo, tanto en el envío como en la recepción, sea óptima, y que se reduzca al mínimo la posibilidad de perder correos o que se corrompan los mensajes antiguos en el servidor.

- **Servicio de terminal remoto para linux (SSH)**: se necesita un acceso por SSH a una máquina Linux para que el personal de ARCOS pueda ejecutar mandatos Linux. Es deseable que en el acceso por SSH de cada cuenta de usuario se tenga acceso a los datos de dicha cuenta.

- **Servicio de DHCP**: se necesita un servidor de DHCP que permita la asignación automática de direcciones IP al personal que tenga un dispositivo móvil con conexión *ethernet* y que desee acceder a la red de ARCOS.

3.3.2. Directrices del sistema

Los servicios ofrecidos por el sistema han de orientarse según las siguientes directrices:

- Fiabilidad de los datos.
  - **Uso de RAID como solución de aumento de fiabilidad de datos**: se necesita utilizar estrategias de RAID para ofrecer fiabilidad de datos, por lo tanto, es necesario buscar un tipo de RAID que soporte tolerancia a fallos.
- Seguridad.
  - **Seguridad de los datos de asignaturas, proyectos y datos personales**: se necesita establecer permisos para mantener la privacidad de los datos más críticos de cada cuenta. Ningún usuario debe tener acceso o posibilidad de obtenerlo para entrar en cuentas no autorizadas.
  - **Separación de clientes/servidores**: es necesario que los usuarios no tengan cuentas locales en los servidores, únicamente los administradores; de esta forma se obliga a los usuarios a trabajar en las máquinas cliente o en una máquina dedicada exclusivamente como servidor SSH.
- Rapidez.
  - **Reunifiación de cuentas de usuario (Windows y Linux)**: es la forma más cómoda para los usuarios, dado que tendrán una única cuenta con un solo nombre de usuario y contraseña, que podrán cambiar en cualquiera de los dos sistemas. Además todos los datos de la cuenta estarán almacenados en un único sitio, con la posibilidad de acceder a ellos de diversas formas.
  - **Administración más sencilla de usuarios**: se necesita realizar la administración de cuentas de la manera más eficiente posible, buscando la comodidad para los administradores y la mayor automatización posible, mediante el uso de *scripts* o herramientas ya desarrolladas.
- Disponibilidad.
  - **Facilidad de recuperación de las maquinas, los servicios y los datos críticos**: se necesita una solución que permita restablecer la operatividad de las máquinas, los servicios y el acceso a los datos más críticos en el menor tiempo posible. Para ello es necesario contemplar estrategias de *backup* con una fácil recuperación de los datos copiados.

- **Alta disponibilidad**: es necesario establecer un medio de alta disponibilidad, para que en caso de caída de los sistemas principales, los servicios más críticos puedan seguir ejecutándose mientras se recupera el sistema completo.
- **Separación física de las máquinas críticas**: la separación física permite establecer medidas de alta disponibilidad y estrategias de *backup* más eficientes. En caso de caída de una de las máquinas físicas, debe existir al menos otra máquina física que continúe ofreciendo la misma funcionalidad que la anterior, al menos de los servicios más críticos.

  En cuanto a la localización física, se desea que las máquinas residan en una sala con una temperatura adecuada para contener servidores, las 24 horas del día durante todo el año. Además debe ser un sitio con acceso restringido, para evitar que cualquier persona por accidente pueda apagar una de las máquinas.

Además hay que tener en cuenta el presupuesto para la realización del proyecto, que ha de ser lo más bajo posible. Por tanto, se va ha realizar un análisis que determine qué se va a utilizar para cubrir las necesidades y reduciendo en la medida de lo posible el presupuesto.

3.4. ¿Que se va a realizar?

En este punto se describe qué se va a realizar finalmente. De este nuevo sistema se pretende que tenga al menos la misma funcionalidad que el anterior sistema de servidores, pero con mejoras que eviten los problemas del antiguo modelo y ofrezcan un mayor rendimiento.

3.4.1. Hardware

La idea de una restructuración de los servidores era inminente, y dado que se contaba con máquinas más potentes, en un principio destinadas a formar *clusters* de cómputo, se decidió utilizar algunas de ellas para crear un nuevo sistema de servidores. El hardware disponible para la realización del proyecto se compone finalmente de las máquinas listadas a continuación, contando con el antiguo *Aguila*, como una máquina más del nuevo sistema:

- **Intel Pentium 4 2.8Ghz con 1GB de memoria RAM**. Viene en formato de *rack* con dimensión 3U. Dispone de 3 discos duros de 250GB de capacidad. En adelante *Daisy*.

- **Intel Core 2 DUO 2,13Ghz con 4GB de memoria RAM**. Viene en formato de *rack* con dimensión 4U. Dispone de 8 discos duros Serial ATA con distintas capacidades de almacenamiento. En adelante *Donald*.
- **AMD K7 ATHLON XP 2000+ con 512MB de memoria RAM**. Viene en formato ATX normal, no es de tipo *rack* y dispone de 4 discos duros ATA de 120GB de capacidad. En adelante *Boyerito*.

3.4.2. Servicios

Los servicios a proporcionar por el nuevo sistema de servidores son los listados en el apartado *¿Que se necesita?* (ver página 70.), y son comparados con el antiguo sistema en el apartado *¿Cómo estaba anteriormente el sistema?* (ver página 66.). En el presente apartado se hace un análisis sobre qué se va a utilizar para ofrecer cada servicio y las razones que argumentan la elección de ese modelo.

- **Autenticación de usuarios, tanto para Windows como para Linux**:

  Dado que se pretende unificar las cuentas de usuario de Windows y Linux ya existentes, se necesita un sistema de autenticación que sirva para los dos sistemas. De los sistemas comentados anteriormente (ver capítulo '*Estado de la cuestión*', Sistemas de autenticación , página 45.), Ldap es el que mejor se adapta a la solución buscada dado que cumple los siguientes puntos:

  - Autenticación de usuarios tanto en Windows como en Linux gracias a *backends* como Samba.
  - Rapidez en las búsquedas dentro del directorio de datos. (El directorio se formara con datos de equipos, grupos y personas)
  - Posibilidad de utilizar comunicaciones cifradas.
  - Permite replicar los datos y recuperarlos de una manera muy sencilla y económica.
  - Posibilidad de utilizar un servidor secundario para realizar balanceo de carga.

  El producto Ldap que mejor se adapta al presupuesto es OpenLdap, una solución Ldap de código abierto y gratuita.

  Por otro lado, MySQL no ofrece el mismo rendimiento para la autenticación de usuarios, además no hay tantas herramientas que soporten la autenticación con MySQL, no sucediendo lo mismo con Ldap. En cuanto a NIS, la seguridad es uno de los factores que desestiman su elección, dado que no contempla cifrados en sus comunicaciones. OpenLdap se muestra como la solución de autenticación más óptima para el proyecto.

	Rapidez búsquedas	Cifrado en las comunicaciones	Soporte en múltiples herramientas (para la autenticación)
OpenLdap	**SI**	**SI**	**SI**
Mysql	NO	SI	NO
NIS	SI	NO	NO

Cuadro 3.1: Comparativa de soluciones de autenticación

- **Almacenamiento para los usuarios y otros datos de los que dispone el área**: Los datos a utilizar deben residir en algún espacio de disco desde el cual se pueda acceder tanto desde sistemas Windows como sistemas Linux. Una posible solución es almacenarlos en un sistema Linux que permite exportar los datos a máquinas Windows y Linux. En el caso de clientes Linux, el espacio de disco se exporta utilizando *NFS* y en el caso de clientes Windows se exporta dicho espacio con *SAMBA*.

  Este modelo permite tener los datos centralizados y posibilita que varios usuarios trabajen a la vez sobre los mismos, sin tener que replicarlos, obteniendo unicidad. La última versión de NFS (la 4), incluye seguridad Kerberos, trabaja con cortafuegos y permite ACLs.

  Como se dispone de una máquina con 8 discos duros Serial Ata, ésta será la encargada del almacenamiento de los datos; además se desea tener fiabilidad en los mismos, por lo tanto hay que utilizar soluciones que aumenten la fiabilidad. Para seguir manteniendo el servicio, a la vez que la fiabilidad de datos ante el mal funcionamiento de uno de los discos, es preferible utilizar un RAID. De los distintos tipos de RAID se necesita uno que permita el fallo de al menos uno de los discos, lo que descarta la utilización de un RAID 0. Del resto de opciones hay que contemplar el tipo de accesos a disco que tendrá el sistema; en éste caso, la mayor parte de accesos serán de escritura y lectura de pequeños ficheros (servicio de correo) y un sistema RAID 5 tiene un rendimiento más pobre que un RAID 1 en ese tipo de accesos.

  Es importante también contemplar la posibilidad de extraer un disco del RAID y poder leer los datos del mismo. Únicamente el RAID1 ofrece esta posibilidad, dado que los discos que forman el espejo, tienen cada uno el mismo sistema de ficheros, que puede leerse como si no se tratase de un RAID. Esto facilita el acceso a los datos en caso de necesidad.

  El sistema RAID1 es el que mejor rendimiento y capacidades ofrece al sistema propuesto, permitiendo el fallo de uno de los discos que lo integran y obteniendo

un óptimo rendimiento en accesos de lectura, y un rendimiento normal, sin ser más costoso, en los accesos de escritura. Como desventaja se reducirá la capacidad que sumaban los 8 discos de los que contaba la máquina a la mitad, pero no se prevé una utilización del tamaño final (contando con el RAID1) superior al 40 %, resultando ser una opción igualmente válida.

	Tolerancia a fallos	Rapidez lectura/escritura (pequeños bloques)	Acceso a los datos con un solo disco
RAID0	NO	SI	NO
RAID1	**SI**	**SI**	**SI**
RAID5	SI	NO	NO

Cuadro 3.2: Comparativa de soluciones RAID

Además de utilizar un sistema RAID1, se desea tener un medio que asegure la posibilidad de aumentar el espacio de almacenamiento de una manera cómoda. La mejor solución es utilizar volúmenes lógicos utilizando los RAID1 disponibles, de esta manera se consiguen dividir los datos entre los volúmenes, permitiendo aumentar el espacio de cada volumen en caso de necesidad.

- **Servicio WEB**: Se busca una solución fiable, rápida y con el menor coste posible. IIS de Microsoft implementa un servidor web entre otras funciones, pero tiene la desventaja de requerir Microsoft Windows como sistema operativo, con su correspondiente coste por licencia.
  Apache es otra solución web que ofrece un óptimo rendimiento, fiabilidad (continuamente salen parches de seguridad) y una gran aceptación como solución de servidor web en la red. Su desventaja es la falta de una interfaz gráfica que permita configurarlo, no obstante, permite una gran flexibilidad en su configuración. Además acepta autenticación a través de Ldap, siendo muy útil para el directorio Ldap que se va a utilizar como solución de autenticación global. Ésto último permitirá a los usuarios utilizar sus cuentas para acceder a determinadas webs o crear cuentas específicas para la autenticación de ciertas páginas. Por tanto, la solución web que se utilizará en el sistema sera Apache.

- **Servicio DNS**:Para el servicio DNS no se seguirán utilizando los subdominios *linux.arcos.inf.uc3m.es* y *win.arcos.inf.uc3m.es*, existirá solo el dominio principal *arcos.inf.uc3m.es*. Se utilizará BIND9 para la resolución de nombres bajo una máquina Linux, lo que permite disponer de un servidor secundario.

  En el caso de los Windows, éstos estarán bajo un dominio definido en Samba, por

lo tanto el servidor DNS a utilizar será el mismo que las máquinas Linux. Además Samba actuará como servidor WINS, facilitando la resolución de nombres en el dominio Samba para Windows.

- **Servicio de correo**:El sistema requiere un servidor SMTP, un servidor POP3, un servidor IMAP y un cliente web de correo, que ya se proporcionaban en *Aguila*. Se seguirá utilizando el mismo servidor SMTP de antes: Postfix, y servidores Pop3 e Imap con opción de soporte SSL para Linux. Además, para evitar correos con *spam* ó virus, se utilizará la solución Postfix+Amavis+Clamav+Spamassasin: Amavis utiliza Clamav (un antivirus) como antivirus para los correos y Spamassasin para filtrar el contenido con *spam*.

  El formato que se utilizaba anteriormente en *Aguila* era *mailbox*, cuyo principal problema residía en la lentitud en los accesos de lectura y escritura del correo, dado que éstos se guardaban en el mismo fichero por cada carpeta existente en el servidor. Además existía el riesgo de que se corrompiera la cabecera de algún correo dentro de un fichero imposibilitando la lectura de los correos siguientes, obligando a editar el fichero manualmente para solucionar el problema.

  La solución que se va a utilizar para evitar estos problemas es el uso del formato *Maildir*, que consiste en tener un fichero por cada correo. Esta solución acelera la lectura y escritura de correos y evita que se impida la lectura de varios correos en caso de que se corrompa un fichero. Postfix permite usar el formato *Maildir*, resultando ser la opción más eficaz para el nuevo sistema.

- **Servicio de terminal remoto para linux**: Se seguirá utilizando SSH como opción para conectarse a un terminal remoto linux para que los usuarios puedan tener una *shell* Linux. En dicho acceso los usuarios tendrán acceso a su directorio *home* de Linux donde aparecerán todos los datos de su cuenta, incluidas las webs (tanto normal como en modo seguro *https*). La máquina que ofrezca el acceso por SSH tendrá solo esta función, llevando un registro exhaustivo de los comandos ejecutados por cada usuario como medida de seguridad, facilitando las operaciones de búsqueda forense en caso de incidencias.

- **Servicio de DHCP**: Se utilizará el servidor Dhcp3, una solución de servidor *dhcp* de código abierto para Linux. Para ello se establecerá un rango de IPs dentro del segmento de ARCOS para la asignación por DHCP.

3.4.3. Uso de máquinas virtuales

Para cubrir las necesidades del nuevo sistema y ofrecer los servicios listados anteriormente del modo que se ha descrito, se ha pensado en una solución basada en máquinas

virtuales.

Ésta solución permite tener varios sistemas ejecutándose, cada uno de los cuales ofrecerá distintos servicios salvaguardando la seguridad de los datos. Además, las medidas de seguridad de cada sistema se adaptarán a los servicios ofrecidos, en lugar de tener que adaptarse a todos a la vez, lo que reduciría la eficacia de las mismas. Por otro lado, ante cualquier tipo de incidencia, es posible seguir ejecutando las máquinas virtuales desde otro equipo preparado para ello (migración de máquinas virtuales), reduciendo el tiempo sin servicio ante cualquier incidencia, obteniendo alta disponibilidad en el sistema.

De los sistemas de virtualización comentados en el capítulo *Estado de la cuestión* (ver página 20.), la solución más óptima es Xen, dado que permite ejecutar máquinas virtuales utilizando el sistema de paravirtualización que se aproxima al rendimiento nativo de una máquina. Xen permite paravirtualización utilizando sistemas operativos modificados, y virtualización completa sobre procesadores con tecnología *Intel-VT* o *AMD-V*. Cómo se dispone de una máquina con procesador *Intel Core 2 Duo* que incorpora la tecnología Intel-VT, se podría instalar un sistema operativo sin modificar utilizando virtualización completa.

No obstante, la paravirtualización ofrece un mejor rendimiento al estar el sistema operativo de la máquina virtual modificado para ejecutarse sobre un *hypervisor* no siendo necesario que éste monitorice todas las instrucciones, sino que los sistemas operativos huésped y anfitrión colaboran en la tarea.

La máquina con más potencia de cómputo y más capacidad de disco, *Donald* (ver página 83, apartado 4.2.1), será la encargada de almacenar y ejecutar las máquinas virtuales. El sistema operativo anfitrión escogido para ésta máquina es Linux, en concreto la distribución Debian Etch (distribución estable de Debian) y todas las máquinas virtuales que se ejecuten estarán basadas en Debian Etch o Debian Sarge (la distribución estable anterior, más consolidada que Etch). La elección del sistema operativo Linux se basa en el elevado coste que supondría una licencia para Windows frente al coste cero que tiene un sistema operativo Linux. Por otro lado, Linux puede ser modificado para Xen, utilizando la paravirtualización en lugar de la virtualización completa que se requeriría para Windows, aumentando el rendimiento de las máquinas virtuales.

3.4.4. Directrices del sistema

A continuación se listan las directrices que debe seguir el sistema y las argumentaciones según las cuales, el sistema propuesto cumple con dichas directrices:

- Fiabilidad de los datos.

- **Uso de RAID como solución de aumento de fiabilidad de datos**: el nuevo sistema cumple con la directriz de fiabilidad de datos con el uso de RAID1 para el almacenamiento en disco. Según se ha comentado en el apartado anterior, RAID1 ofrece tolerancia a fallos, rapidez en el acceso a los datos con un solo disco del RAID y un acceso óptimo en las lecturas y escrituras.

- Seguridad.

  - **Seguridad de los datos de asignaturas, proyectos y datos personales**: Los datos han de estar en un sistema de ficheros con permisos (*ext3* de Linux cumple este punto) de tal forma que cada usuario solo pueda acceder a sus datos, o a aquellos cuyos permisos le concedan acceso. Además, la exportación de datos a las distintas máquinas virtuales tendrá que ser restrictiva en cuanto a usuarios privilegiados como *root* (uso de *root_squash*).
  - **Separación de clientes/servidores**: Una de las ventajas que supondrá tener máquinas virtuales es que se destinará una de ellas como servidor SSH para los usuarios. El motivo de esta decisión es el evitar que cualquier usuario pueda, de manera intencionada o no, bloquear una máquina crítica, como sucedería en el caso de que pudiesen ejecutar instrucciones directamente sobre un servidor. Al restringir el espacio de ejecución a una máquina virtual, únicamente podrán bloquear los recursos de la misma en el peor de los casos, sin afectar al resto de las máquinas virtuales y el sistema anfitrión.

    Únicamente los administradores tendrán acceso por SSH a los distintos servidores.

- Rapidez.

  - **Reunificación de cuentas de usuario (Windows y Linux)**: Mediante la autenticación a través de un directorio Ldap, ámbos sistemas (Windows y Linux) podrán utilizar la misma información de cuentas. Las máquinas con Linux obtendrán la información del directorio Ldap para cualquier operación relacionada con los usuarios; esto es posible por los múltiples módulos existentes para programas que necesitan de autenticación o búsqueda de información sobre un usuario, como Apache, o las librerías *pam* de Linux.

    En el caso de Windows, la autenticación se realizará a través de Samba, que posee módulos de autenticación a través de Ldap. Una vez que el usuario se haya autenticado en un cliente Windows, aparecerá como una unidad más del sistema su directorio *home* de Linux, es decir, la información de su cuenta de usuario. De esta forma, el usuario trabajará con la opción de guardar todos

sus datos en la nueva unidad montada y así tenerlos accesibles desde cualquier máquina.

- **Administración más sencilla de usuarios**: Al contar con una única base de datos de usuarios, se reducen las dificultades y el trabajo de administración de usuarios. Además Ldap cuenta con varios interfaces gráficos *opensource* de administración vía web, facilitando las labores de administración más básicas. Por otro lado, para las tareas que requieran de *scripts*, existen también herramientas de Ldap en Linux que realizan de manera más sencilla las operaciones sobre el directorio.

- Disponibilidad.

  - **Facilidad de recuperación de las maquinas, los servicios y los datos críticos**: La recuperación de máquinas virtuales es muy económica, dado que se realizarán copias de seguridad de los sistemas raíz de todas, así como una copia de cada sistema raíz dentro de un fichero imagen, listo para ejecutarse en el servidor de respaldo.

    Los servicios básicos también contarán con copias de respaldo, para recuperar su configuración y los datos de que dispongan. Según el tipo de servicio, se realizará copia de seguridad lógica, física o ambas. Por ejemplo, la base de datos MySQL puede tener una copia física: ficheros de configuración y directorio donde se guardan los ficheros con las tablas; y una copia lógica: volcado de todas las bases de datos en un fichero de texto.

    Respecto a los datos críticos (los datos de usuarios y otros proyectos de ARCOS), residen originalmente en *Donald*, que contará con discos en RAID1, permitiendo el fallo de un disco por cada RAID. Además se realizará una copia de seguridad en *Daisy* (con objeto de tener sincronizados los datos en el servidor de respaldo) y por último, una copia de seguridad en *Boyerito*, cuya localización física esta alejada del armario donde residirán *Donald* y *Daisy*. Ante cualquier incidente en una de las localizaciones, se tendrá copia de seguridad en la otra.

  - **Alta disponibilidad**: La máquina Intel Pentium 4 2.8Ghz con 1GB de memoria RAM, *Daisy*(ver página 86, apartado 4.2.2), se utilizará como servidor de alta disponibilidad para *Donald*. Ésta máquina contendrá las mismas máquinas virtuales y los datos de los usuarios repartidos en sus discos duros (solo los de los usuarios, sin contar con las copias de seguridad y otros datos). Cuando *Donald* falle o haya que realizar cualquier tipo de mantenimiento sobre él, entrará en funcionamiento *Daisy*, ofreciendo el mismo servicio a los usuarios temporalmente, mientras la primera máquina vuelve a estar operativa. Para

llevar a cabo este proceso, se deberá realizar una sincronización al menos diaria del contenido de las máquinas virtuales y los datos de los usuarios.

Hay que tener en cuenta que los datos de los usuarios están siempre en continuo cambio, y las máquinas virtuales utilizan dichos datos para los distintos servicios que ofrecen. Si se tuviesen que ejecutar las máquinas virtuales en el servidor de respaldo, los datos de los usuarios a los que éstas acceden tendrían que ser actualizados primero, o anular el acceso a dichos datos (dado que son una copia con un día de diferencia en el peor de los casos), permitiendo solo utilizar los servicios básicos, como DNS, web, etc, hasta que se recupere la última instancia de los datos de usuario.

En la figura 3.1 se puede observar el esquema descrito anteriormente. *Donald* es la encargada de ejecutar las máquinas virtuales y de servir los datos de disco; por otro lado, *Daisy* esta en espera por si *Donald* falla o es desconectado, para cargar la réplica de las máquinas virtuales y los datos de usuario previamente sincronizados. Como se ha descrito anteriormente, si no pueden sincronizarse los datos de *Donald* a *Daisy*, éste último ejecutará las máquinas virtuales sin acceso a los datos, ejecutando solo los servicios básicos.

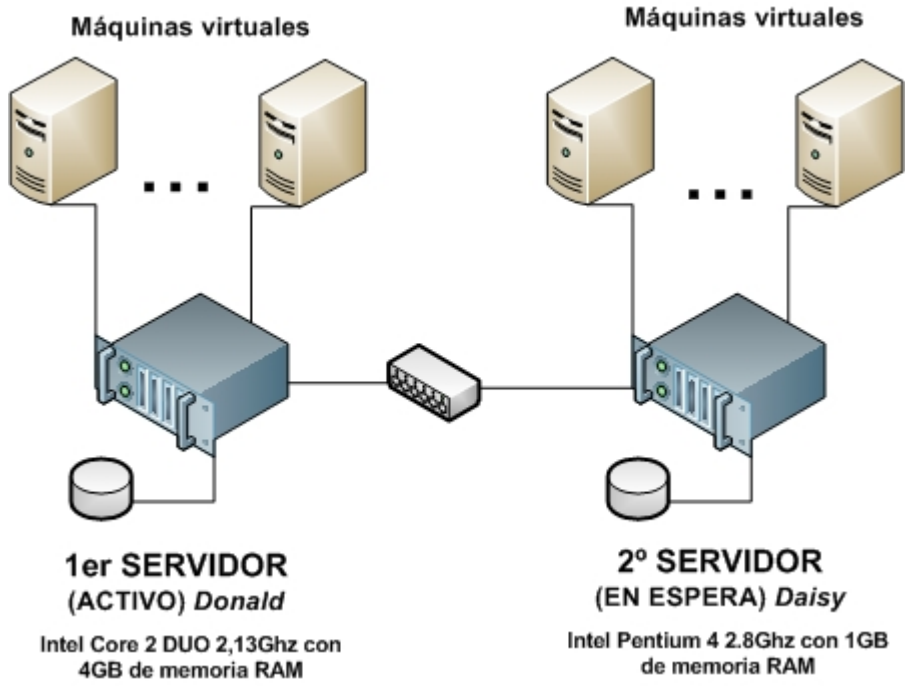


Figura 3.1: Servidores de máquinas virtuales

- **Separación física de las máquinas críticas**: Como se ha descrito antes con *Donald* y *Daisy*, ambas máquinas son físicas, de tal forma que la carga de las máquinas virtuales y los datos, las pueden realizar ambas manteniendo la alta disponibilidad.

  Además es importante señalar la localización de las máquinas que componen el sistema: tanto *Donald* como *Daisy* residirán en el mismo armario situado en el centro de cálculo de la Universidad. Esta sala cuenta con una refrigeración adecuada para los servidores las 24 horas del día, durante todo el año; además, el armario donde están introducidas tiene ventilación propia, disipando el calor fuera del mismo.

  En el caso de *Boyerito*, éste reside en el laboratorio, que no tiene las mismas condiciones atmosféricas y está al alcance de cualquier persona, pero no puede instalarse en el centro de calculo al no ser una máquina en forma de *rack*. La ventaja es que está en una localización física muy alejada de *Donald* y *Daisy*, y como su función es la de realizar copias de seguridad, asegura que siempre exista una copia de respaldo en caso de catástrofe en una de las dos localizaciones.

Capítulo 4

Diseño

4.1. Introducción

Como se ha visto anteriormente en el capítulo de análisis, existirán tres máquinas físicas y varias máquinas virtuales. El presente capítulo se divide en tres secciones: dedicación de las máquinas físicas, dedicación de las máquinas virtuales, y un resumen general del diseño de todo el sistema. En cada una de las secciones se describirán los objetivos de cada máquina y cómo se realizan.

4.2. Dedicación de las máquinas físicas

Esta es la sección donde se describe el objetivo de cada una de las máquinas físicas y cómo se realiza. Las máquinas físicas del sistema son: *Donald*, *Daisy* y *Boyerito*. Según el esquema visto en el capítulo de análisis (ver figura 4.1), *Donald* y *Daisy* realizarán la carga de las máquinas virtuales, contendrán los datos de las cuentas de usuario y llevarán a cabo una sincronización entre ellas, para mantener un sistema de alta disponibilidad. Por otro lado *Boyerito* servirá para realizar copias de seguridad de los datos que mantienen las otras dos máquinas.

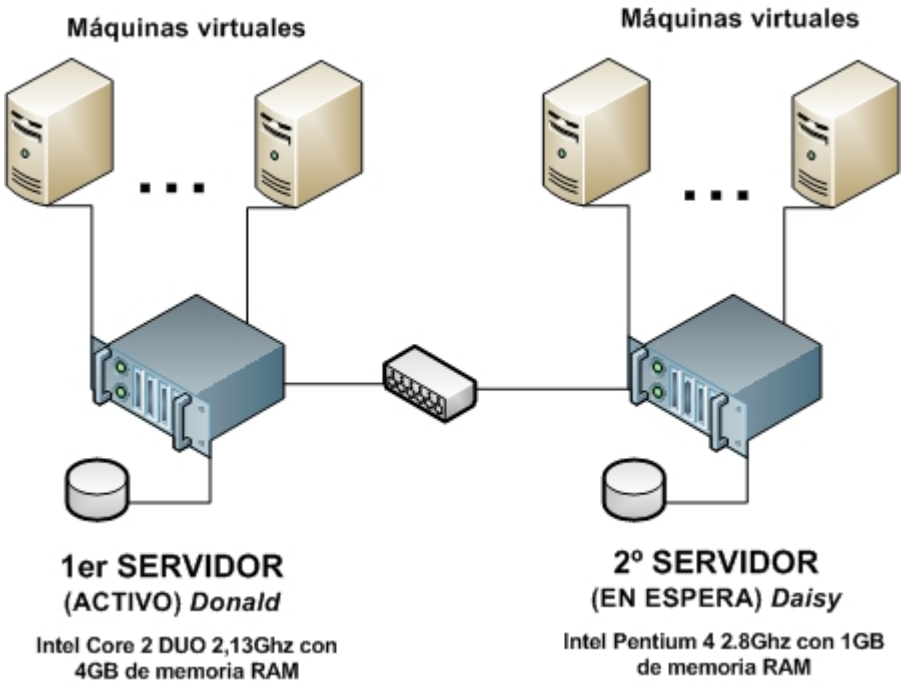


Figura 4.1: Máquinas físicas del sistema

4.2.1. *Donald*

Localizada dentro de un armario en el centro de cálculo de la Universidad, *Donald* será la máquina física principal de todo el sistema de *ARCOS*. Los pasos relativos a la instalación del sistema operativo *Debian Linux*, optimizado para realizar paravirtualización mediante *Xen*, vienen como anexo del presente documento (ver página 220.). Las funciones de *Donald* serán:

- **Carga de las máquinas virtuales**: el sistema operativo será modificado para poder ejecutar máquinas virtuales utilizando la versión 3 de *Xen*. Por defecto, *Donald* será la encargada de ejecutar dichas máquinas, que se iniciarán a la vez que *Donald* por un orden establecido según la funcionalidad de cada una, y que se verá más adelante (ver página 115.).

- **Almacenamiento de los datos**: los datos de *ARCOS* (cuentas de usuarios, asignaturas, proyectos, etc.) serán almacenados en *Donald* para su uso por las distintas máquinas virtuales. Habrá una máquina virtual dedicada a la exportación de dichos datos por NFS al resto de máquinas, dado que solo una máquina a la vez, puede acceder al dispositivo donde residen los datos. El objetivo de *Donald* es el de mostrar un dispositivo físico a la máquina virtual que exportará los datos para que ésta lo

use como disco duro normal.

Para el almacenamiento de los datos, se utilizarán volúmenes lógicos, consiguiendo varios espacios de almacenamiento separados en función del tipo de datos que se guarde. En concreto se utilizarán 3 volúmenes lógicos según la siguiente clasificación de datos a almacenar:

1. **Espacio para las cuentas de usuario**: en este volumen lógico se guardarán los datos relativos a cada cuenta de usuario, es decir, el directorio *home* del mismo en el sistema Linux. En este mismo espacio, y como parte de los datos de cada cuenta, se guardaran también las páginas web del usuario (modo normal y modo seguro) y los correos electrónicos.
2. **Espacio de *backup* para cada usuario**: cada cuenta de usuario dispondrá de un directorio dedicado al almacenamiento de los datos más críticos. Este volumen lógico se dedica al almacenamiento de dicha información, proporcionando una separación física de éstos datos respecto al los del resto de la cuenta, aumentando la fiabilidad del almacenamiento.
3. **Espacio para *backups* del propio sistema**: el último volumen lógico se utilizará para almacenar las copias de seguridad de las máquinas y servicios que componen el sistema, así como otros datos y máquinas antiguas de ARCOS.

La descripción de cómo están estructurados los datos dentro de cada volumen lógico se verá más adelante.

Donald cuenta con 8 discos duros Serial Ata conectados formando 5 *RAID1* (dos discos por cada *RAID*, salvo el último *RAID*, que aprovecha parte de otros dos discos ya utilizados); la figura 4.2 ilustra esta configuración.

Los volúmenes lógicos están formados a partir de los primeros 4 *RAID1* según muestra la figura 4.3: por cada uno de los cuatro *RAID1*, se define un volumen físico, después se agrupan los 4 volúmenes físicos en un grupo de volúmenes y por último del grupo de volúmenes, se extrae el espacio disponible para formar los 3 volúmenes lógicos comentados.

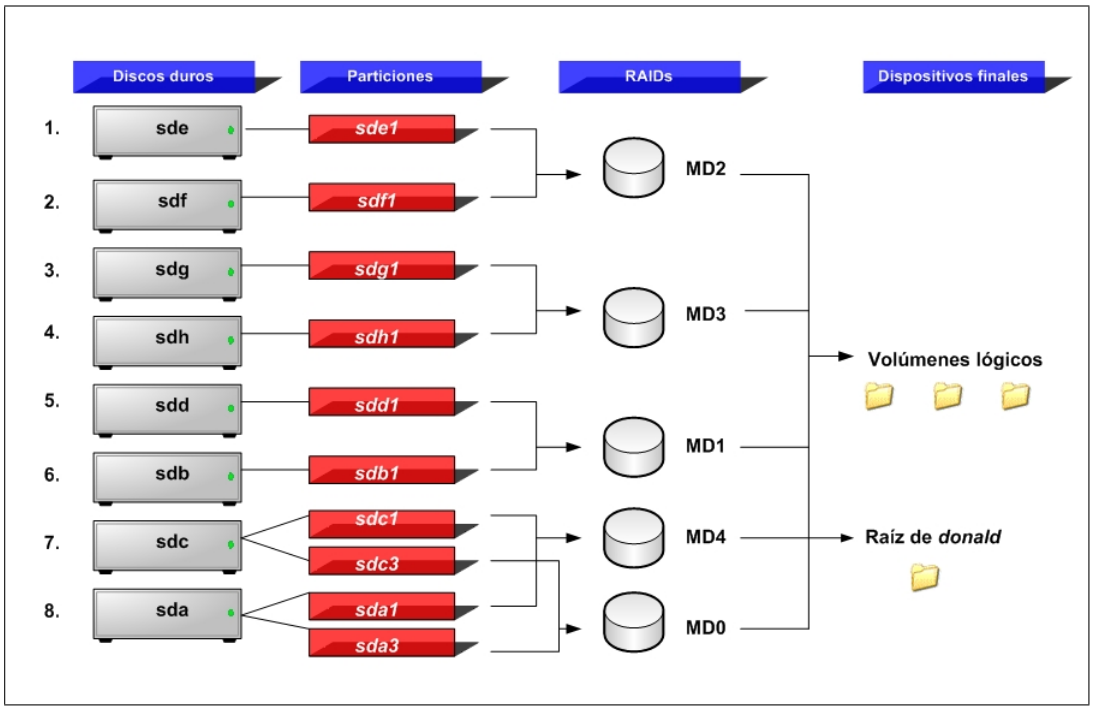


Figura 4.2: Configuración de discos en *Donald*

- **Sincronización con Daisy**: los volúmenes lógicos no podrán ser montados por *Donald* ya que éstos los utilizará una de las máquinas virtuales como se describe al comienzo del punto anterior. Para realizar la sincronización, *Donald* accederá a los volúmenes lógicos montándolos por NFS a través de la máquina virtual que tiene acceso a los mismos. Dado que ésta máquina virtual se ejecutará desde el propio *Donald*, la comunicación por NFS se realizará internamente (por el interfaz *loopback*), con una pérdida de rendimiento aceptable, en comparación con el acceso directo a los volúmenes lógicos.

  Se utilizarán *scripts* para la sincronización de los volúmenes lógicos de *Donald* a *Daisy*. Éstos *scripts* se ejecutarán a través del *crontab* de *Donald* en horario nocturno y todos los dias. La primera sincronización será la más costosa, dado que se tiene que realizar una transferencia íntegra de los datos almacenados en cada uno de los tres volúmenes lógicos. No obstante, a partir de la segunda transferencia y en adelante, los cambios serán pocos, y se realizará la sincronización en pocos minutos.

  Los *scripts* de sincronización seguirán los siguientes pasos:

  - Se comprobará la conexión con *Daisy*, en caso de fallo se ejecutará el último punto.

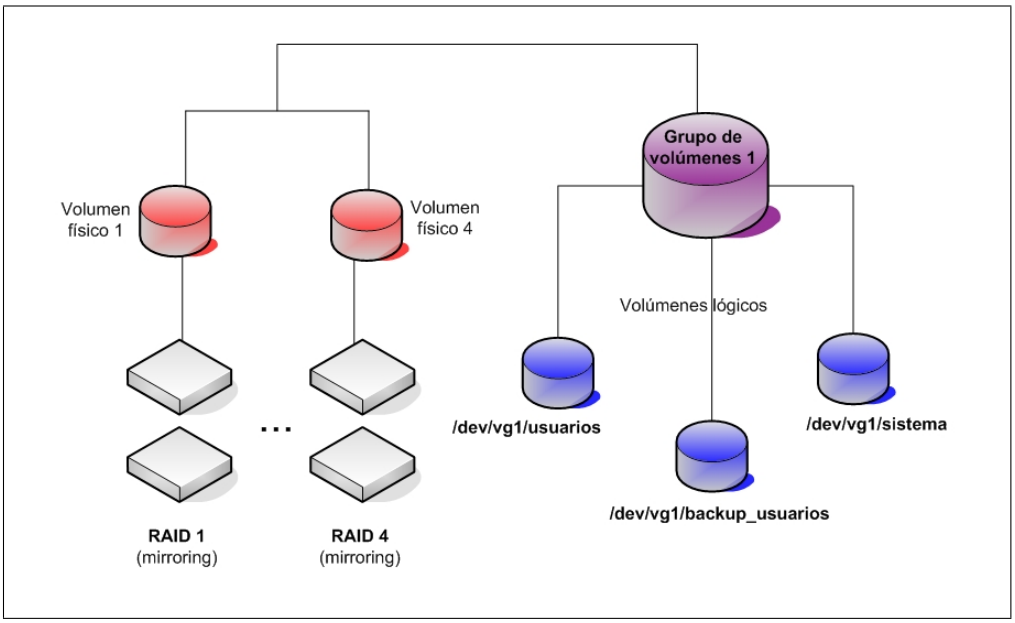


Figura 4.3: Volúmenes lógicos en *Donald*

- Montará en *Daisy* el volúmen lógico correspondiente al *script* que se ejecute (*/dev/vg1/usuarios*, */dev/vg1/backup_usuarios* o */dev/vg1/sistema*).
- Montará en *Donald* también el volúmen lógico correspondiente por NFS a través de la máquina virtual que tiene acceso a los volúmenes lógicos.
- Realizará un *rsync* desde *Donald* a *Daisy* utilizando los directorios montados en ambas máquinas.
- Desmontará el volúmen lógico en *Donald* (montado por NFS).
- Desmontará el volúmen lógico en *daisy*.
- Mandará un *email* a los administradores con el tiempo de inicio, el tiempo final, y estadísticas de la transferencia realizada. En caso de que no se haya realizado, también mandará un *email* avisando del fallo.

4.2.2. *Daisy*

Localizada en el mismo armario que *Donald*, *Daisy* será la máquina física que permitirá tener un sistema de alta disponibilidad, al ser el respaldo de *Donald*. Los pasos relativos a la instalación del sistema operativo Linux Debian Etch, optimizado al igual que *Donald* para realizar paravirtualización mediante *Xen*, vienen como anexo del presente documento (ver página 8.2). *Daisy* tendrá las mismas funciones que *Donald* pero como sistema de respaldo:

- **Carga de las máquinas virtuales**: el sistema operativo será modificado para poder ejecutar máquinas virtuales utilizando la versión 3 de *Xen*. Por defecto *Donald* será la encargada de ejecutar dichas máquinas, pero en caso de fallo o de necesidad, Daisy las ejecutará en su lugar. Como *Daisy* es una máquina menos potente, el rendimiento de las máquinas virtuales se verá reducido, no obstante, éste servicio por parte de *Daisy* deber ser solo provisional hasta que *Donald* esté operativa.

- **Almacenamiento de los datos**: en la sección de *Donald* sobre almacenamiento de los datos (ver página 83.), se explicaba que los datos de *ARCOS* (cuentas de usuario, asignaturas, proyectos, etc.), se almacenaban en el propio *Donald*. También se explicaba que existiría una máquina virtual que tendría acceso a dichos datos y los exportaría a las demás por NFS. Y por último, se comentaba que el objetivo de *Donald* respecto al almacenamiento de datos, era el mostrar un dispositivo físico a la máquina virtual que exportase los datos para que ésta lo tratase como un disco duro normal.

  En el caso de *Daisy*, al tratarse de una máquina de respaldo, deberá ofrecer la misma funcionalidad que *Donald*, pero de manera temporal. Por ello, también contará con un espacio de almacenamiento para los datos de *ARCOS*. Además, *Daisy* tendrá que ejecutar las máquinas virtuales cuando no puedan ejecutarse en *Donald*, incluyendo la máquina virtual que accede a los datos de *ARCOS*, debiendo tener acceso igualmente a ellos en *Daisy*. Por tanto, *Daisy* deberá mostrar también un dispositivo físico de almacenamiento a la máquina virtual que exporta los datos de *ARCOS*, para que ésta lo use como un disco duro normal.

  Para el almacenamiento, *Daisy* tiene cuatro discos duros de *250GB ata100* que se dividirán en las mimas 4 particiones:

  1. ***hda1, hdb1, hdc1 y hdd1***: tendrán 25 *Gigas* de espacio y estarán destinadas a formar un *RAID10* para el sistema raíz.
  2. ***hda2, hdb2, hdc2 y hdd2***: tendrán 1 *Giga* de espacio y estarán destinadas a servir de espacio *swap*.
  3. ***hda3, hdb3, hdc3 y hdd3***: tendrán 214 *Gigas* de espacio y estarán destinadas a formar un *RAID5* para el almacenamiento de los datos de *ARCOS*.
  4. ***hda3, hdb3, hdc3 y hdd3***: tendrán 50 *Mb* de espacio y estarán destinadas a formar un RAID1 con dos discos de reserva (*spare disk*) para el directorio */boot*.

  Con las particiones *hda1, hdb1,hdc1 y hdd1* se creará un *RAID10*. La razón por la que se crea este tipo de RAID, es que se necesita un espacio de almacenamiento de al menos 40GB para alojar el sistema raíz (incluyendo las imágenes de las máquinas

virtuales). Dado que se tienen particiones de 25Gigas, se pueden crear dos RAID1, obteniendo dos espacios de almacenamiento separados de 25 Gigas cada uno y con tolerancia a fallos, y juntarlos mediante un RAID0. El RAID10 realiza esto automáticamente.

Las particiones *hda1, hdb2, hdc2 y hdd2* se dejarán como espacio de intercambio (*swap*). Por otro lado, con las particiones *hda3, hdb3, hdc3 y hdd4* se creará un *RAID5* que tendrá 591,36 *Gigas* de espacio final de almacenamiento, suficientes para albergar todo el contenido de los datos de *ARCOS*.

La razón de que no se utilice un *RAID1* que es más óptimo para éste sistema (como ya se analizó en el capítulo de análisis, ver página 74.), es que no se dispone de los suficientes discos duros en *Daisy* como para formar un *RAID1* con el espacio necesario para albergar los datos de *ARCOS* suponiendo un cierto margen de aumento. Con el *RAID5* se tendrán 591,26 *Gigas*, espacio suficiente según las previsiones. En el caso de que aumentase la utilización del espacio en *Donald* (éste cuenta con 1 *Terabyte* aproximadamente), se almacenarían únicamente los datos de las cuentas de usuario, para que se puedan utilizar temporalmente, si entra *Daisy* en modo respaldo de *Donald*. La configuración de las particiones y dispositivos finales en *Daisy* se pueden ver en las figuras 4.4 y 4.5 .

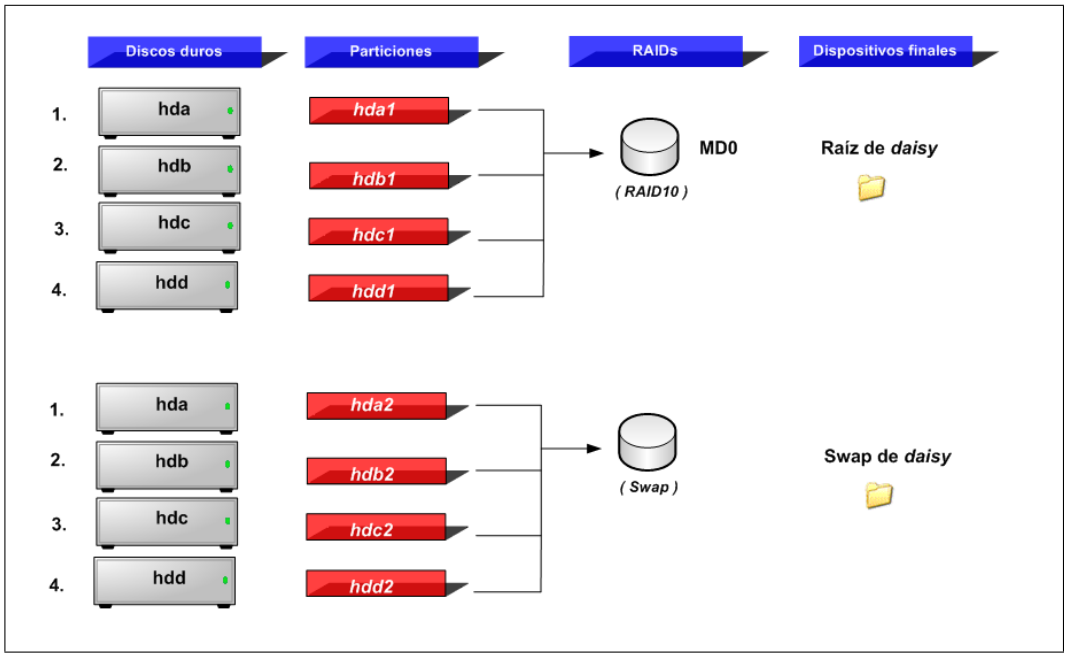


Figura 4.4: Configuración de discos en *Daisy* (1ª parte)

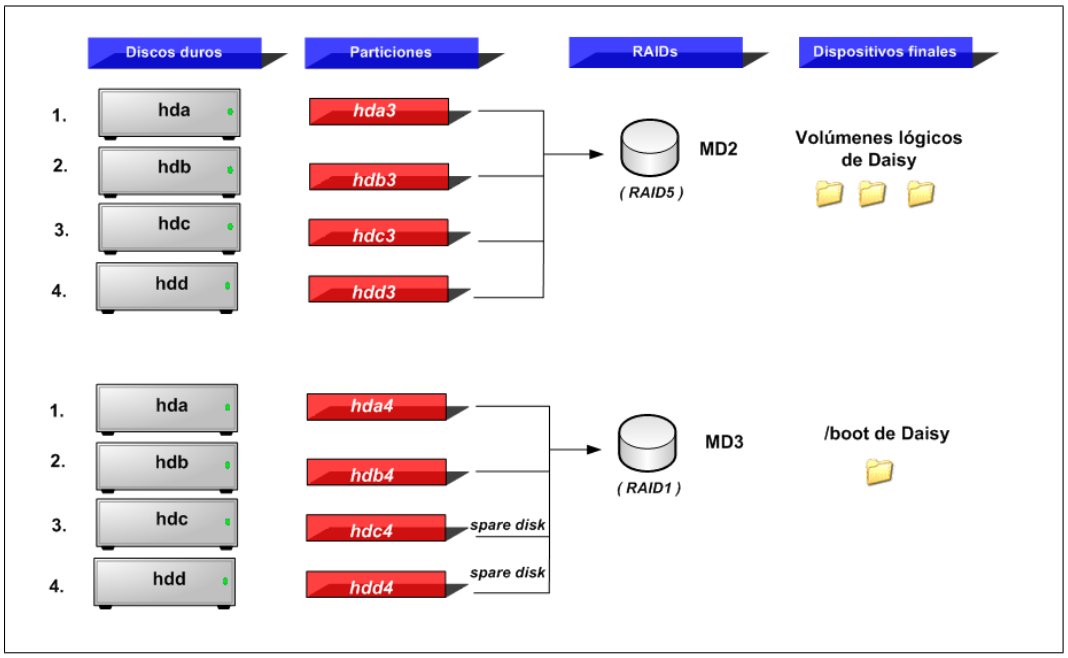


Figura 4.5: Configuración de discos en *Daisy* (2ª parte)

Al igual que *Donald*, *Daisy* dispondrá de tres volúmenes lógicos con la misma denominación que en *Donald*: */dev/vg1/usuarios*, */dev/ vg1/ backup_usuarios* y */dev/ vg1/ sistema*. Éstos volúmenes se formarán con el *RAID5* creado de la siguiente forma: se marcará el *RAID5* como volumen físico y con éste se creará un grupo de volúmenes. Del grupo de volúmenes se obtendrá el espacio para los volúmenes lógicos. La razón de que se utilicen volúmenes lógicos en *Daisy* en lugar del *RAID5* directamente es para permitir la posibilidad de añadir más discos y aumentar las particiones creadas para cada volumen lógico. Además de ésta forma, se pueden utilizar los mismos ficheros de configuración de cada máquina virtual que existen en *Donald*, dado que están configurados para utilizar volúmenes lógicos con la misma nomenclatura. La figura 4.6 muestra el esquema de volúmenes lógicos en *Daisy*.

- **Sincronización con *Donald***: como se comentaba en la sección de *Donald* sobre la sincronización con *Daisy* (ver página 85.), *Donald* accedía a los volúmenes lógicos montándolos por *NFS* a través de la máquina virtual que sí tenía acceso a los mismos. Cuando *Daisy* entre en modo respaldo de *Donald*, ejecutará las máquinas virtuales, incluyendo aquella que accede a los volúmenes lógicos y exporta los datos por *NFS*. Ésta máquina virtual se ejecutará en *Daisy* cuando tenga los datos de los volúmenes lógicos totalmente sincronizados con los de *Donald*, sino, se ejecutarán solo el resto de las máquinas virtuales, hasta que se tengan sincronizados los datos.

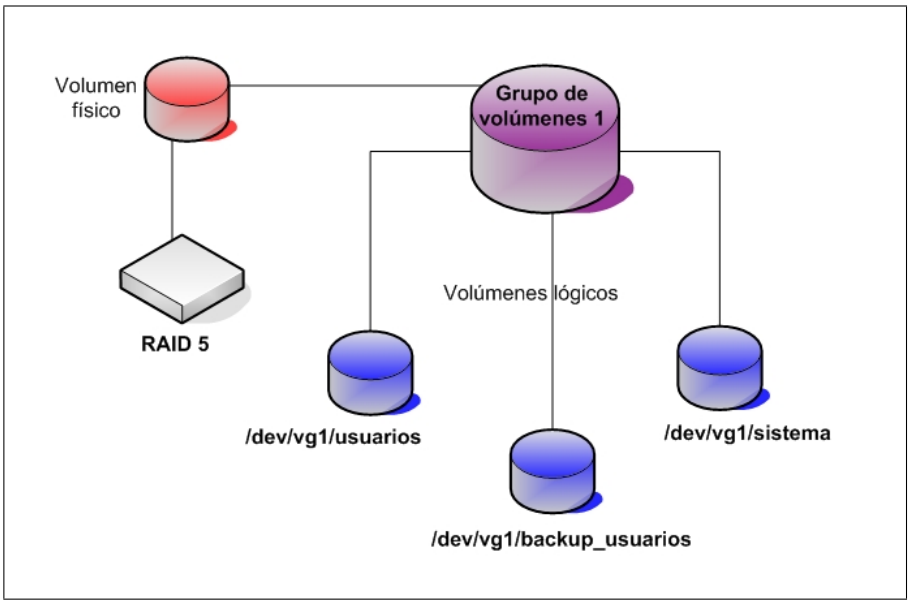


Figura 4.6: Volúmenes lógicos en *Daisy*

Mientras *Donald* no este operativa, se estarán utilizando los volúmenes lógicos de *Daisy* y los datos almacenados en ellos irán cambiando. Cuando *Donald* vuelva a estar operativa, se deberá realizar una sincronización inversa, es decir, sincronizar los datos de los volúmenes lógicos de *Daisy* a los volúmenes lógicos de *Donald*. Para ello se deberá parar la máquina virtual que accede a los volúmenes lógicos; el resto pueden seguir funcionando (ofreciendo los servicios mínimos mientras que se realiza la sincronización y vuelven a ejecutarse en *Donald*). *Daisy* realizará la sincronización hacia *Donald* con un *script* por cada volúmen lógico; cada *script* seguirá los siguientes pasos:

- Comprobará la conexión con *Donald*, en caso de fallo el *script* irá al último punto.
- Montará en *Donald* el volúmen lógico correspondiente al *script* que se este utilizando. El volúmen lógico se montará en *Donald* directamente, no a través de *NFS* como en otros casos.
- Montará también en *Daisy* el volúmen lógico correspondiente. Como la máquina virtual que accedía a los mismos estará apagada, no habrá ninguna incidencia.
- Realizará un *rsync* de *Daisy* a *Donald* utilizando los directorios correspondientes a los volúmenes lógicos montados.
- Desmontará de *Donald* el volúmen lógico.

- Desmontará de *Daisy* el volúmen lógico.
- Mandará un *email* a los administradores con el tiempo de inicio, el tiempo final, y estadísticas de la transferencia realizada. En caso de que no se haya realizado, también mandará un *email* avisando del fallo.

4.2.3. *Boyerito*

Localizada en el laboratorio de *ARCOS*, *Boyerito* será la máquina física dedicada a contener una copia de seguridad de los datos que se almacenarán en los volúmenes lógicos de *Donald* y *Daisy*. Anteriormente era el servidor principal de *ARCOS*, *Aguila*, pero en el nuevo sistema se ha decidido reutilizarla. Tendrá el sistema operativo Debian Linux Etch y dispondrá de varios discos duros para realizar la copia de seguridad.

Necesidad de una máquina de *backup* aislada

Uno de los inconvenientes de la proximidad física de *Donald* y *Daisy* (están en el mismo armario dentro del centro de cálculo), es que ante cualquier tipo de incidente en la localización en la que residen, que dejase inoperativas las máquinas, dejará a *ARCOS* sin todos sus datos. Para prevenir una situación así, la mejor opción es realizar un *backup* de todos los datos que se almacenan en *Donald* y *Daisy* en otra localización física distinta, y para ello se pensó en reutilizar el antiguo servidor *Aguila*. Al estar *Aguila* en el laboratorio de *ARCOS*, se dispondría de un respaldo de todos los datos en otra localización física, aumentando la fiabilidad de todo el sistema. *Aguila*, por tanto,se reinstalará y renombrará como *Boyerito* para realizar la función de respaldo de los datos.

Copias de seguridad en *Boyerito*

Para el almacenamiento de la copia de los datos de *Donald* y *Daisy*, *Boyerito* dispondrá de 4 discos duros de *120GB ATA100*. Se necesita tener un espacio de almacenamiento de al menos *300GB* según las previsiones de lo que ocuparán los datos de *ARCOS* y copias de seguridad de las máquinas y servicios que componen el sistema. Además se necesita que dicho espacio de almacenamiento tenga tolerancia a fallos, por lo tanto se usará una estrategia de *RAID*.

Las estrategias que se pueden dar son *RAID1* y *RAID5*. Al contar con solo 4 discos duros de *120GB*, se debe descartar el *RAID1*, dado que se podrían formar dos *RAID* de este tipo y unirlos con un *RAID0*, o tratarlos por separado, pero en cualquier caso, la capacidad total del sistema no superaría los *240GB*, insuficientes para almacenar todos los datos de *ARCOS*. La opción que queda es la de formar un *RAID5* con los 4 discos duros. También hay que contar con espacio para el sistema raiz, asi que se pensará en crear varias particiones en cada disco, dejando una partición de al menos *110GB* para

formar el *RAID5*. Con 4 particiones de *110GB* para formar el *RAID5* se dispondrá de un espacio de almacenamiento de casi *300GB*, suficientes para almacenar todos los datos de las cuentas de usuario y parte de las copias de seguridad de las máquinas y servicios del sistema.

El esquema final de particiones de *Boyerito*, ilustrado en la figura 4.7, quedará asi:

- ***hda1 y hdc1***: tendrán 9 *Gigas* de espacio y estarán destinadas a formar un *RAID1* para el sistema raíz.
- ***hda2 y hdc2***: tendrán 1 *Giga* de espacio y estarán destinadas a servir de espacio *swap*.
- ***hda3, hdb3, hdc3 y hdd3***: tendrán 110 *Gigas* de espacio y estarán destinadas a formar un *RAID5* para almacenar el *backup* de los datos de *ARCOS*.

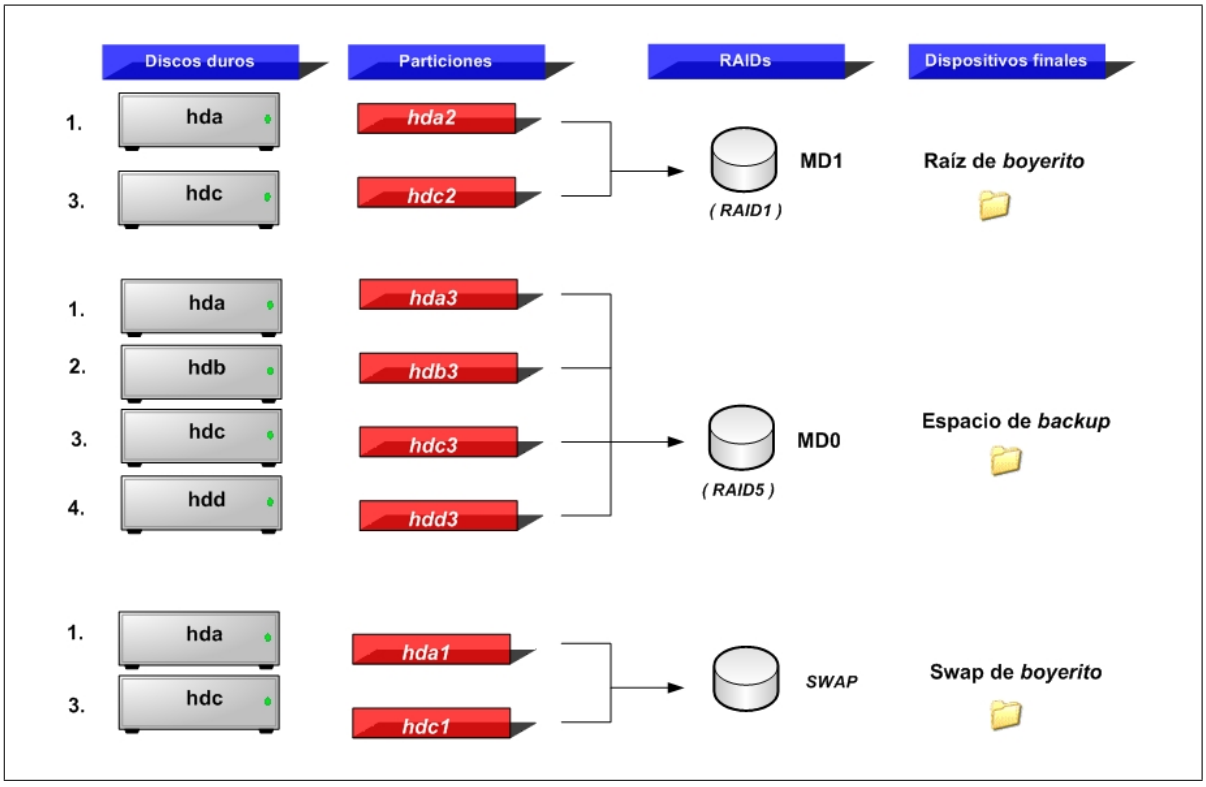


Figura 4.7: Configuración de discos en *Boyerito*

El horario en que se realizará la copia de seguridad en *Boyerito* será nocturno, dado que la utilización del sistema y el acceso a los datos será reducido por parte de los usuarios. El sistema estará por tanto con poca carga, y el aprovechamiento de las horas

nocturnas para realizar las copias de seguridad no supondrán un descenso del rendimiento del sistema, dada la poca utilización del mismo en horas nocturnas.

Boyerito estará programada para encenderse por las noches y esperar a que se realice la copia de seguridad en ella. Una vez que la copia de seguridad se realice, se apagará para evitar tener en funcionamiento sus discos duros más de lo necesario, aprovechando al máximo su vida útil.

La copia de seguridad se realizará desde la máquina virtual que tiene acceso a los volúmenes lógicos (tanto de *Donald* como de *Daisy*). Esta máquina ejecutará 3 *scripts*, que copiaran los datos de cada uno de los volúmenes lógicos a distintos directorios dentro del *RAID5* de *Boyerito*. Cada uno de esos *scripts* seguirá los siguientes pasos:

- Comprobará la conexión con *Boyerito*, en caso de fallo el *script* irá al último punto.
- Montará en *Boyerito* el *RAID5* en el directorio */mnt/md0*.
- Realizará un *rsync* del directorio de la máquina virtual donde esté montado el volumen lógico correspondiente al *script*, al directorio destino en *Boyerito* donde se almacenarán esos datos. Si es necesario, se excluirán algunos directorios por falta de espacio en el directorio destino (backups de las máquinas del sistema y de otras máquinas antiguas, menos relevantes).
- Desmontará de *Boyerito* el *RAID5*.
- Mandará un *email* a los administradores con el tiempo de inicio, el tiempo final, y estadísticas de la transferencia realizada. En caso de que no se haya realizado, también mandará un *email* avisando del fallo.

Una vez ejecutados los 3 *scripts* correspondientes a los volúmenes lógicos */dev/vg1/usuarios*, */dev/vg1/backup_usuarios* y */dev/vg1/sistema*, se mandará la orden de apagado a *boyerito* desde la máquina virtual que ejecuta dichos *scripts*.

4.3. Dedicación de las máquinas virtuales

En esta sección se describe la funcionalidad de cada máquina virtual que aparecerá en el sistema. Se utilizarán 4 máquinas virtuales: una para servicios externos, una para autenticación, otra para almacenamiento y *backup*, y la última como terminal remoto (SSH). La figura 4.8 muestra la ejecución de las máquinas virtuales desde *Donald*.

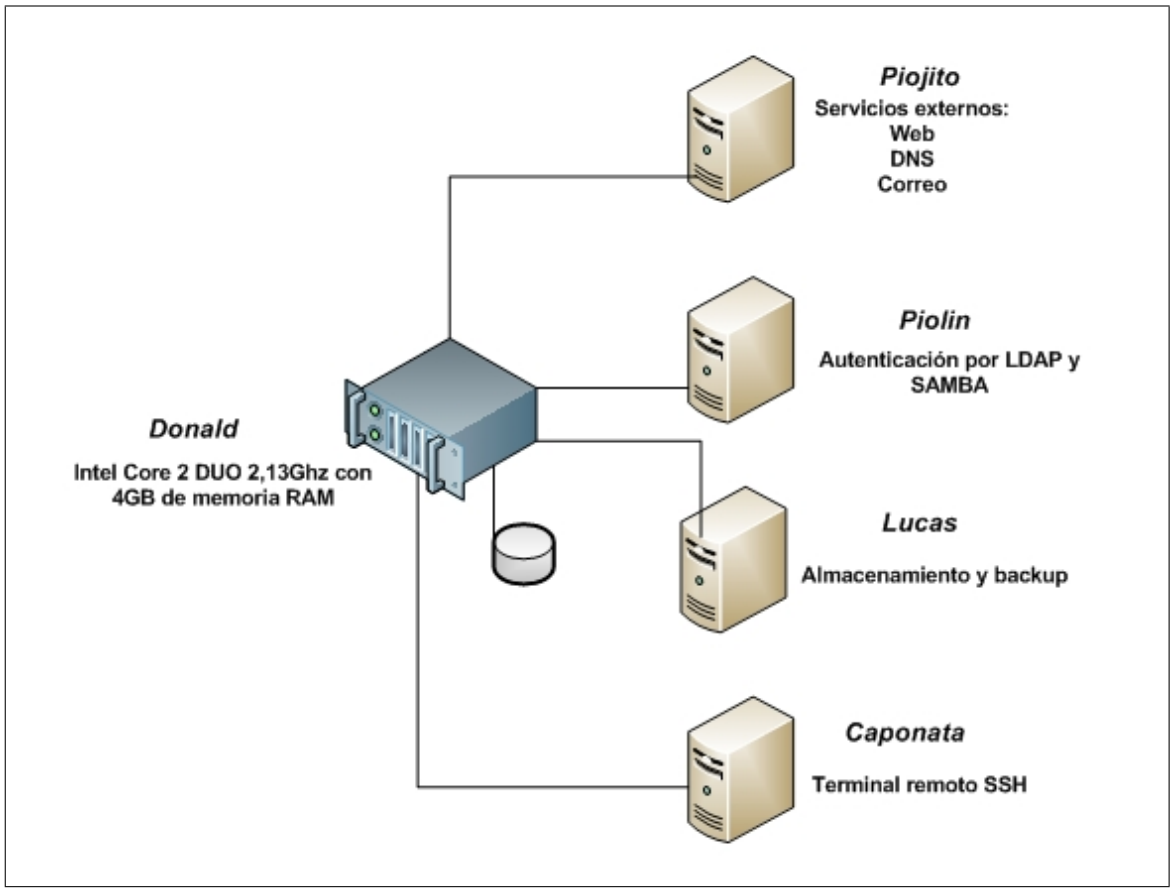


Figura 4.8: Máquinas virtuales en *Donald*

4.3.1. *Piolin*, autenticación

Piolin será la máquina virtual encargada de la autenticación. Su función principal será la gestión de las cuentas de usuario, ofreciendo varios servicios que realizan operaciones sobre las mismas.
Un resumen de los servicios que ofrecerá es el siguiente:

- Servidor Ldap de cuentas de usuario.
- Administración de usuarios.
- Autenticación bajo Ldap para Linux.
- Autenticación bajo Ldap para Windows usando Samba.

Piolin como máquina virtual independiente

La razón de que exista una máquina virtual con solo este servicio es la protección de las cuentas de usuario y la información de los mismas. *Piolin* montará el espacio de almacenamiento del sistema para las cuentas de usuario (el volumen lógico */dev/vg1/usuarios* y */dev/vg1/backup_usuarios*) a través de *Lucas* (encargado de exportar los volúmenes lógicos). Montará estos directorios con permisos *no_root_squash* (el usuario root tendrá permisos totales sobre el sistema montado), dado que el usuario *root* es el encargado de la creación de los mismos y no debe tener restricciones de acceso. Al restringir la máquina a dar solo servicio de autenticación y denegar el acceso de cualquier usuario que no sea el administrador, se protegen los datos de los usuarios. Los únicos accesos a ésta máquina serán de otros sistemas que necesiten autenticación utilizando el servidor Ldap, y de los administradores.

Ldap en *Piolin*

Para llevar a cabo los servicios que ofrecerá *Piolin*, se utilizarán las siguientes herramientas: Openldap, Smbldap-Utils y Samba. Openldap se encargará de mantener la base de datos de las cuentas y sobre ella se realizarán las operaciones de consulta, modificación, borrado o creación. La base de datos Ldap, utilizará los *schemas*(ver página 51.) de Posix y Samba, de tal forma que cada usuario tendrá información relativa a un sistema Linux y un sistema Windows. Un ejemplo de entrada Ldap de una cuenta de usuario es el siguiente:

```
dn: uid=folcina,ou=People,dc=arcos,dc=inf.uc3m.es
objectClass: posixAccount,inetOrgPerson,shadowAccount,sambaSamAccount
uid: folcina
cn: Francisco Olcina Grande
sn: Grande
uidNumber: 3021
gidNumber: 3000
gecos: Francisco Olcina Grande,,,
shadowMax: 99999
shadowWarning: 7
sambaSID: S-1-5-21-3492619381-3135118558-3133272105-7042
loginShell: /bin/bash
homeDirectory: /mnt/home/becarios/folcina
sambaProfilePath: \\lucas\profiles
sambaLogonScript: netlogon.bat
sambaHomeDrive: Z:
sambaHomePath: \\lucas\homes
sambaPasswordHistory: 00000000000000000000000000000000
```

```
shadowLastChange: 13459
sambaKickoffTime: 1230764399
sambaLMPassword: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
sambaNTPassword: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
sambaPwdCanChange: 1176989672
sambaPwdMustChange: 2147483647
sambaPwdLastSet: 1176989672
sambaAcctFlags: [U          ]
userPassword: {SMD5}xxxxxxxxxxxxxxxxxxxxxxxxxxxx
mail: folcina@arcos.inf.uc3m.es
```

Como se puede observar, contiene información que utiliza Linux: *uid, gid, homeDirectory, userPassword, loginShell*; e información que utiliza Windows: *sambaProfilePath, sambaHomePath, sambaLMPassword* etc.

Las herramientas que contiene el paquete Openldap para la manipulación de entradas dentro del directorio son poco intuitivas, por lo que se ha optado por utilizar el paquete *smbldap-utils*. Este paquete contiene herramientas más automatizadas para el tratamiento de entradas en un directorio Ldap que utilizen los *schemas* Posix y Samba. Los *scripts* del paquete *smbldap-utils* son los siguientes:

- ***smbldap-useradd***: añade un usuario nuevo.
- ***smbldap-userdel***: borra un usuario.
- ***smbldap-usermod***: modifica cualquier atributo de un usuario.
- ***smbldap-usershow***: muestra la información sobre un usuario.
- ***smbldap-userinfo***: modifica los campos de un usuario preguntando por cada uno de ellos.
- ***smbldap-groupadd***: añade un grupo nuevo.
- ***smbldap-groupdel***: borra un grupo existente.
- ***smbldap-groupmod***: modifica un grupo.
- ***smbldap-groupshow***: muestra la información sobre un grupo.
- ***smbldap-passwd***: establece o modifica la contraseña de un usuario.
- ***smbldap-populate***: crea los grupos y usuarios iniciales que necesita un dominio Windows.

Los grupos que existirán finalmente serán los del dominio Windows, creados con el comando *smbldap-populate*, y los necesarios para *ARCOS*:

- **Docencia**
- **Profesores**
- **Becarios**
- **Proyectos**
- **Alumnos**
- **Otros**
- **Masters**
- **Invitados**

Cada persona ,proyecto o asignatura de *ARCOS*, contará con una cuenta de usuario dentro de los grupos listados anteriormente.

Autenticación bajo ldap

Por cada cuenta de usuario se crearán una serie de directorios que incluye el directorio *home* de la cuenta. Estos directorios se montarán en el sistema de *Piolin* desde *Lucas* por NFS, con permisos *no_root_squash* como se ha comentado anteriormente. Se desarrollará un *script* que utilice las herramientas *smbldap-useradd* y *smbldap-passwd* para automatizar el proceso de creación de los directorios.

Un sistema Linux utiliza los números *uid* y *gid* para asignar un propietario a cada fichero, enlace simbólico o directorio. Estos números se obtienen del directorio Ldap mediante las librerías *libnss-ldap*. Por otro lado, un sistema Linux autentica a los usuarios mediante las librerias *pam*. Utilizando el módulo *libpam-ldap* es posible autenticar usuarios existentes en el directorio Ldap además de los que aparezcan en los ficheros *group*, *passwd* y *shadow*.

En Windows la autenticación se realiza a través de Samba. *Piolin* también ejecutará un servidor Samba configurado como controlador primario de dominio. Esto permite a un ordenador con Windows instalado, unirlo al dominio Samba, como si de un dominio Active Directory se tratase. Samba realiza la búsqueda de usuarios a través del directorio *Ldap* y además puede añadir, borrar, modificar o consultar la información de las cuentas de usuario; dentro de su configuración se utilizan los *scripts* del paquete *smbldap-utils* para llevar a cabo las operaciones mencionadas.

Recursos del servidor Samba en *Piolin*

Existirán tres rutas predefinidas en Samba para el acceso a ciertos recursos. Dichas rutas apuntarán al servidor Samba *disco*, que es el servidor Samba que se ejecuta en *Lucas*, analizado más adelante. Los recursos ofrecidos son:

- ***logon script***: es la ruta que indica donde reside el *script* de inicio. Este *script* es un fichero de procesamiento por lotes que Windows ejecutará cada vez que el usuario inicie la sesión. La ruta configurada es: *disco**netlogon**netlogon.bat*.
- ***logon home***: es la ruta donde Windows buscará el directorio *home* de esa cuenta de usuario. La ruta configurada es: *disco**homes*.
- ***logon path***: es la ruta donde Windows buscará el perfil móvil del usuario. En este sistema no se quieren tener perfiles móviles, por lo que está desactivada.

En resúmen, las máquinas que utilicen Linux se autenticarán directamente contra el servidor Openldap, mientras que las máquinas que utilicen Windows lo harán a través de Samba configurado como *backend* del servidor Openldap. La figura 4.9 muestra este diseño.

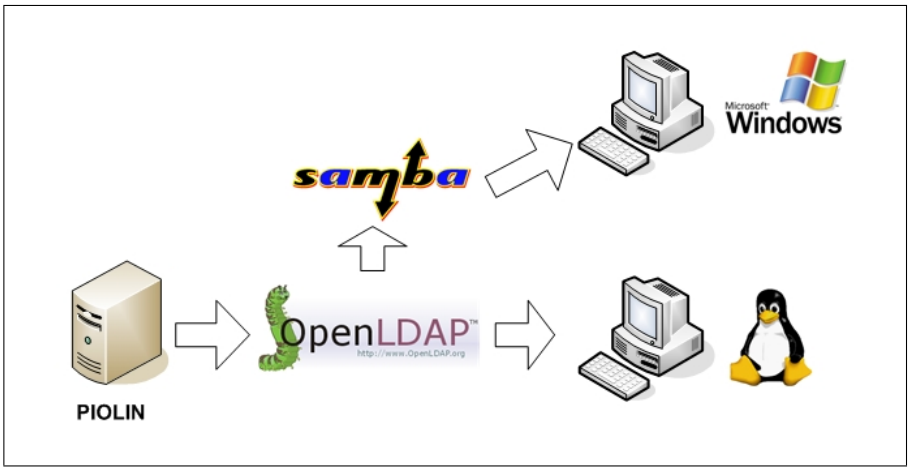


Figura 4.9: Autenticación a través de *Piolin*

4.3.2. *Lucas*, almacenamiento y backup

Lucas será la máquina virtual encargada de exportar el espacio de almacenamiento de las cuentas de usuario al resto de máquinas y de realizar las copias de seguridad. Para ello contará con acceso a los volúmenes lógicos donde se almacenan los datos, realizará la exportación de los mismos por NFS y Samba, y ejecutará mediante el *crontab* de Linux, *scripts* para realizar los *backups*.

Lucas como máquina virtual independiente

Lucas se constituirá como una máquina virtual con los servicios comentados para obtener mayor seguridad en los datos del sistema y obtener transparencia y rapidez al realizar las copias de seguridad. La seguridad de los datos se dará al restringir el acceso a la máquina salvo a los administradores, o mediante el uso de NFS o Samba, impidiendo a cualquier usuario el acceso a otros servicios o a ejecutar comandos dentro de la máquina. Los permisos tanto de NFS como de Samba estarán adaptados para ofrecer la mayor seguridad al sistema. Más adelante se comentará la configuración usada.

Respecto a la función de realizar las copias de seguridad, se obtendrá rapidez al ser *Lucas* máquina virtual que las realiza, puesto que tendrá un acceso directo a los volúmenes lógicos, en concreto al volumen lógico */dev/vg1/sistema*, donde se almacenará la información de *backup*. En cuanto a la transparencia, tanto *Donald* como *Daisy* (las máquinas físicas donde se ejecutarán las virtuales) mostrarán a *Lucas* los volúmenes lógicos como si se tratasen de discos duros, por lo que *Lucas* seguirá realizando sus tareas de *backup* con independencia de la máquina física donde se esté ejecutando, añadiendo transparencia al sistema.

Utilización de los volúmenes lógicos de *Donald* y *Daisy* por *Lucas*

El acceso a los volúmenes lógicos se realizará de la siguiente forma: *Donald* y *Daisy* tendrán configurados éstos volúmenes como */dev/vg1/usuarios*, */dev/vg1/backup_usuarios* y */dev/vg1/sistema*, además en el fichero de configuración de *Lucas* que ambas máquinas tendrán (dado que se trata de una máquina virtual), aparecerán las siguientes líneas:

```
disk    = [ 'file:[ruta de la máquina virtual]/lucas.img,sda1,w',
            'file:[ruta de la máquina virtual]/swap.img,sda2,w',
            'phy:/dev/vg1/sistema,sda3,w',
            'phy:/dev/vg1/usuarios,sda4,w',
            'phy:/dev/vg1/backup_usuarios,sda5,w' ]
```

El parámetro *disk* en el fichero de configuración de una máquina Xen indica los dispositivos de la máquina anfitrión que se usarán en la máquina virtual y de qué forma. En este caso el fichero *lucas.img* contendrá el sistema raíz de *Lucas* y aparecerá como *sda1*, el fichero *swap.img* se utilizará como espacio de *swap* y aparecerá como *sda2*, y los volúmenes lógicos de *Donald* o *Daisy*, aparecerán en *Lucas* como dispositivos *sda3*, *sda4* y *sda5*, obteniendo la transparencia antes indicada.

Los dispositivos *sda3*, *sda4* y *sda5* en *Lucas*, se montarán como */mnt/sistema*, */mnt/usuarios* y */mnt/backup_usuarios* respectivamente. Dentro de */mnt/usuarios* aparecerán los siguientes directorios:

- **home**.
- **mail**.
- **web**.

La idea es que cada cuenta de usuario tenga separada la información relativa a su *home*, su correo y sus webs en directorios distintos, y el sistema se adapte a ésta configuración. Además cada cuenta posee un directorio extra donde guardar la información más crítica: el directorio de *BACKUP*. Estos directorios son los que se almacenan en */mnt/backup_usuarios*.

El directorio */mnt/backup_usuarios* contendrá una carpeta por cada cuenta de usuario existente, con la información del directorio *BACKUP* que se almacene. Los permisos serán del tipo 700, permitiendo únicamente al usuario propietario de cada carpeta el acceso a la misma.

Por último, en el directorio */mnt/sistema* existirán las carpetas máquinas y servicios: en la carpeta máquinas se realizarán las copias de seguridad de cada máquina que compone el sistema, y en la carpeta servicios se realizarán las copias de seguridad de cada servicio crítico del sistema. Además, en */mnt/sistema* se guardarán otras carpetas que contendrán datos y máquinas antiguas de *ARCOS* de las que se quiere mantener una copia de seguridad.

Configuración de los directorios de las cuentas de usuario

Cada cuenta de usuario tendrá 4 rutas distintas de la siguiente forma:

1. ***Home***: es la ruta del directorio home de la cuenta, cuya dirección es */mnt/home/[nombre del grupo]/[nombre de la cuenta]*.
2. ***Mail***: es la ruta del directorio *Maildir* (de correo) de la cuenta, cuya dirección es */mnt/mail/[nombre del grupo]/[nombre de la cuenta]*.
3. ***Web***: es la ruta donde se guardan las webs de cada cuenta (los directorios *public_html* y *private_html*), cuya dirección es */mnt/web/[nombre del grupo]/[nombre de la cuenta]*.
4. ***Backup***: es la ruta del directorio donde almacenar los datos críticos de la cuenta, cuya dirección es */mnt/backup/[nombre de la cuenta]*.

Dentro del *home* de cada cuenta habrá enlaces simbólicos a los directorios de web y *backup*, según este esquema:

```
public_html -> /mnt/web/[nombre del grupo]/[nombre de la cuenta]/public_html
private_html -> /mnt/web/[nombre del grupo]/[nombre de la cuenta]/private_html
BACKUP -> /mnt/backup/[nombre de la cuenta]
```

En *Lucas*, será necesario instalar la librería *libnss-ldap* para que el sistema obtenga la información de cada cuenta de usuario del directorio Ldap, y así saber a quien pertenecen los permisos de cada fichero o directorio. Como las rutas de cada cuenta de usuario difieren del esquema de directorios de *Lucas* (*/mnt/home/[nombre del grupo]/[nombre de la cuenta]* para un usuario, frente a */mnt/usuarios/home/[nombre del grupo]/[nombre de la cuenta]* que sería la ruta en *Lucas*), se crearán enlaces simbólicos para que el esquema del directorio *mnt* quede de la siguiente forma:

```
#enlace simbólico# backup -> backup_usuarios
#directorio#       backup_usuarios
#enlace simbólico# home -> usuarios/home
#enlace simbólico# mail -> usuarios/mail
#directorio#       sistema
#directorio#       usuarios
#enlace simbólico# web -> usuarios/web
```

De esta forma el directorio home de un usuario tiene una ruta válida dentro de *Lucas*. Esta operación se realiza para la exportación de directorios por Samba, ya que es necesario que cuando Samba acceda al los directorios de una cuenta, encuentre una ruta válida.

Servidor Samba en *Lucas*

El servidor Samba de *Lucas* no actúa como controlador primario de dominio, como es el caso de *Piolin*, sino como servidor de recursos de disco. En concreto ofrece dos recursos, el recurso *homes* y el recurso *BACKUP* correspondientes a los mismos directorios de cada cuenta. El nombre del servidor Samba instalado en *Lucas* es '*disco*', por tanto, los recursos a los que puede acceder un usuario son $\backslash\backslash disco\backslash homes$ y $\backslash\backslash disco\backslash BACKUP$

Cuando un usuario accede a dichos recursos en Windows se le pregunta por su nombre de usuario y contraseña, a no ser que inicie sesión a través del dominio Samba (con *Piolin* como controlador primario de dominio), en cuyo caso se ejecutará el *script netlogon.bat*, cuyo contenido es el siguiente:

```
NET USE H: \\DISCO\BACKUP
NET USE Z: \\DISCO\HOMES
```

Cuando se ejecuta el *script netlogon.bat* al iniciar sesión en el dominio, al usuario le aparecerán dos unidades de red: *H* y *Z*, que se corresponden con el directorio *BACKUP* y el directorio *home* de su cuenta.

Servidor NFS en *Lucas*

Respecto a la exportación por NFS, ésta se realizará de la siguiente forma por cada máquina:

- ***Piolin***: es el servidor de autenticación y desde donde se realizan las tareas de gestión de usuarios, por tanto se necesita acceso a los directorios *home*, *mail*, *web* y *BACKUP* de las cuentas. Como esta máquina no permite acceso a ningún usuario salvo el administrador, y además, necesita crear, borrar y modificar directorios, la exportación tendrá permisos de *no_root_squash* (permite al usuario *root* en la máquina destino tener permisos como tal).

- ***Piojito***: es la máquina que ofrecerá servicios externos, pero únicamente necesita los directorios *web* y *mail* de las cuentas. Se exportarán dichos directorios pero con permisos *root_squash*, dado que el usuario *root* de *Piojito* no debe tener acceso a dichos directorios.

- ***Caponata***: es la máquina que servirá como terminal remoto SSH. Los usuarios tendrán acceso a la máquina y necesitarán también acceso a sus directorios *home*, *mail*, *web* y *BACKUP*. *Lucas* exportará dichos directorios a *Caponata*, pero con permisos de *root_squash*, para minimizar los riesgos de seguridad.

- ***Donald* y *Daisy***: son las máquinas físicas desde las cuales se ejecutarán las virtuales (incluyendo a *lucas*). Para realizar la sincronización de la información que contienen los volúmenes lógicos entre *Donald* y *Daisy*, será necesario acceder a los mismos por parte de ambas máquinas. Como el acceso a los volúmenes lógicos lo realiza *Lucas* (como si de unidades de disco se tratasen), tendrá que exportarlos a la máquina desde donde se estén ejecutando las máquinas virtuales en ese momento, para realizar la sincronización a la otra máquina física. Por tanto *Lucas* exportará a *Donald* y a *Daisy* los directorios */mnt/usuarios*, */mnt/backup_usuarios* y */mnt/sistema*, con permisos de *no_root_squash*, dado que necesitarán que el usuario *root* pueda acceder a todos los directorios para realizar la sincronización.

- ***Resto de máquinas***: las máquinas que necesiten importar los directorios *home*, *mail*, *web* y *BACKUP*, serán aquellas que residan en el laboratorio y tengan Linux instalado. Estas máquinas estarán destinadas a que los alumnos de proyecto fin de carrera realicen los mismos en Linux. Lucas exportará tales directorios salvo el de *BACKUP* y únicamente para el grupo de alumnos, es decir, los directorios: */mnt/home/alumnos*, */mnt/mail/alumnos* y */mnt/web/alumnos*. Estas medidas se toman para que ningún alumno, que realice un proyecto fin de carrera, consiga privilegios en la máquina con Linux que importa tales directorios, y entre en cuentas con información más crítica.

Copias de seguridad en *Lucas*: introducción

Por último, otra de las funciones de *Lucas* será la de realizar las copias de seguridad dentro de */mnt/sistema* (directorio donde se monta realmente el volúmen lógico */dev/vg1/sistema*). Así como las copias de seguridad de todos los directorios (*/mnt/sistema*, */mnt/usuarios* y */mnt/backup_usuarios*) a *Boyerito*.

Copias de seguridad en *Lucas*: *backup* a *Boyerito*

Para las copias de seguridad a *Boyerito*, existirán tres *scripts*, uno por cada directorio a realizar el *backup*. Cada uno de estos *scripts* realizará lo siguiente:

- Comprobará la conexión con *Boyerito*, en caso de que falle irá al último punto.
- Montará en *Boyerito* el directorio */mnt/md0*. (El *RAID5* que tiene integrado).
- Realizará un *rsync* del directorio de *Lucas* del cual se quiere hacer *backup* (*usuarios*, *backup_usuarios* o *sistema*) a */mnt/md0* de *Boyerito*.
- Desmontará el directorio */mnt/md0* de *Boyerito*.
- Mandará un *email* a los administradores con el tiempo de inicio, el tiempo final, y estadísticas de la transferencia realizada. En caso de que no se haya realizado, también mandará un *email* avisando del fallo.

Copias de seguridad en *Lucas*: introducción a *backups* del sistema

Respecto a las copias de seguridad del sistema, se utilizará el directorio */mnt/sistema* de *Lucas*. Las copias serán de dos tipos: copias de seguridad de los sistemas raiz de las máquinas (tanto físicas como virtuales) y copias de seguridad de los datos de los servicios más importantes. Para ello se utilizarán dos subdirectorios de */mnt/sistema*: el directorio máquinas y el directorio servicios. El resto de directorios que aparecen en */mnt/sistema* contienen *backups* de otros datos y máquinas antiguas de *ARCOS*.

Copias de seguridad en *Lucas*: *backup* de las máquinas del sistema

Se distinguen por tanto, respecto a las copias de seguridad del sistema, las copias de las máquinas y las copias de los servicios. Para las copias de las máquinas se realizará un *backup* completo y un *backup* diferencial. Se utilizará el directorio */mnt/ sistema/ maquinas/ [nombre de la máquina]/ raiz* para guardar el *backup* completo y el directorio */mnt/ sistema/ maquinas/ [nombre de la máquina]/[fecha del backup]* para guardar los datos diferenciales respecto a la copia completa. Se creará un *script* que realice las copias de los sistemas raíz de cada máquina y que tenga como parámetro la máquina de la que realizar el *backup*. Este *script* seguirá estos pasos:

- Comprobará la conexión con la máquina de la que realizar el *backup*, en adelante maquina origen.
- Montará en la máquina origen el directorio */mnt/raiz*, cuyo fichero *fstab* estará preparado para montar el raíz del sistema en dicho directorio.
- Realizará un *rsync* diferencial desde la máquina origen al directorio de Lucas correspondiente: */mnt/sistema/maquinas/[nombre de la máquina]/*. Por lo tanto se realizará el *backup* completo en el subdirectorio *raiz* y el diferencial en un subdirectorio que se creará según la fecha.
- Desmontará el directorio */mnt/raiz* de la máquina origen.
- Mandará un *email* a los administradores con el tiempo de inicio, el tiempo final, y estadísticas de la transferencia realizada. En caso de que no se haya realizado, también mandará un *email* avisando del fallo.

Las máquinas de las cuales se realizará una copia de su sistema raiz son: *Donald*, *Daisy*, *Piojito*, *Piolin*, *Lucas* y *Caponata*.

Copias de seguridad en *Lucas*: *backup* de los servicios del sistema

En cuanto a las copias de seguridad de los servicios, estás se realizarán dentro de los directorios */mnt/sistema/servicios/[nombre del servicio]*. Hay dos tipos de copias de seguridad para los servicios, las copias físicas y las copias lógicas. Las copias físicas son aquellas que se realizan de los directorios que contienen los datos de los servicios, y las copias lógicas son volcados a ficheros de texto de la información que manejan los servicios. Habrá un *script* por cada uno de los servicios a realizar copia de seguridad, que son los siguientes:

- **DNS**: el servicio *DNS* se realizará desde *Piojito*. De este servicio se realizará una copia física: se utilizará el comando *rsync* para realizar una copia completa dentro del directorio */mnt/sistema/ servicios/ dns/ bind* y copias diferenciales en */mnt/sistema/ servicios/ dns/ [fecha del backup]*.
- **Ldap**: el servicio *Ldap* se ejecutará desde *piolin*. De este servicio se realizará una copia física y otra lógica. Para la copia física se comprimirá el directorio */var/lib/slapd* que contiene los ficheros que utiliza el servicio Ldap, en un fichero dentro de */mnt/sistema/ servicios/ ldap*, con el siguiente nombre: '*fisico_[fecha]_[hora].bz2*'. Para la copia lógica se utilizará el comando *slapcat* dentro de *Piolin*, guardando la salida en un fichero. El comando realiza por la salida estándar un volcado de toda la información del directorio en formato *ldif*. El fichero generado se guardará en */mnt/ sistema/ servicios/ ldap* con el nombre: '*logico_[fecha]_[hora].ldif*'.

- **MySQL**: el servicio MySQL se ejecutará desde *Piojito*. De este servicio se realizará una copia física y otra lógica. Para la copia física se comprimirá el directorio */var/lib/mysql* que contiene los ficheros que utiliza el servicio MySQL, en un fichero dentro de */mnt/sistema/servicios/mysql*, con el siguiente nombre: '*fisico_[fecha]_[hora].bz2*'. Para la copia lógica se utilizará el comando *mysqldump* dentro de *Piojito*, guardando la salida en un fichero. El comando realiza por la salida estándar un volcado de todas las bases de datos contenidas en el gestor MySQL; además realiza el volcado como si de un *script sql* se tratase, listando todos los comandos necesarios para borrar primero las tablas, crearlas de nuevo e introducir todas las tuplas. El fichero generado se guardará en */mnt/sistema/servicios/mysql* con el nombre: '*logico_[fecha]_[hora].sql*'.

La figura 4.10 ilustra las máquinas y directorios involucrados en las copias de seguridad.

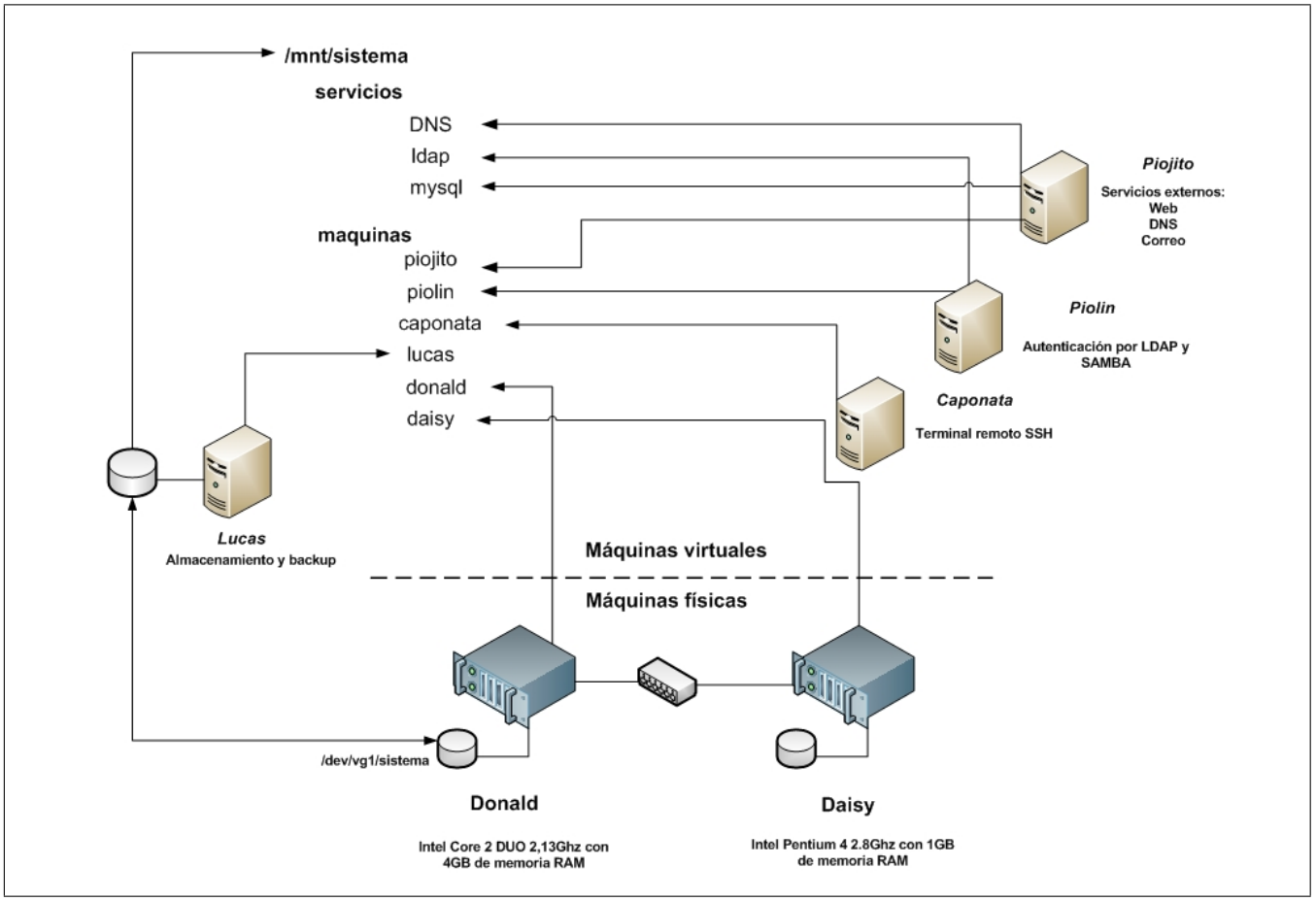


Figura 4.10: Copias de seguridad en *lucas*

Copias de seguridad en *Lucas*: *script* de borrado

Uno de los problemas que conlleva el realizar *backups* diferenciales es la proliferación infinita de directorios. Cada vez que se realiza un *backup*, se crea un nuevo directorio que contiene los datos diferenciales, aumentando el espacio de disco destinado a guardar los *backups* sin ningún control. Para evitar esto, existirá un *script* que realiza una limpieza por cada directorio de *backup* (ya sea de una máquina o un servicio), que borre los ficheros más antiguos, dejando siempre el número de ficheros deseado por directorio. Este *script* recibirá como parámetro la máquina o el servicio del que se quiere realizar la limpieza y el número de directorios o ficheros a conservar.

Copias de seguridad en *Lucas*: horarios

Las copias de seguridad se realizarán por la noche, que es el horario donde menos se utiliza el sistema y menos carga de trabajo tiene. Se programará la ejecución de los *scripts* utilizando el *crontab* de *Lucas*, según el siguiente formato:

```
# backup de la máquina xxxxx
min hora * * * [script de backup de máquinas] [maquina xxxx] &> /dev/null

# borrado de los directorios de la máquina xxxxx
min hora * * * [script de borrado de directorios] [maquina xxxx] &> /dev/null

# backup del servicio xxxxx
min hora * * * [script de backup de servicio xxxx] &> /dev/null

# borrado de los directorios del servicio xxxxx
min hora * * * [script de borrado de directorios] [maquina xxxx] &> /dev/null

#### BACKUP DE LUCAS EN BOYERITO #####
min hora * * * [script de backup de /mnt/usuarios de lucas a boyerito] \
      &> /dev/null
min hora * * * [script de backup de /mnt/backup_usuarios de lucas a boyerito] \
      &> /dev/null
min hora * * * [script de backup de /mnt/sistema de lucas a boyerito] \
      &> /dev/null ; ssh boyerito halt
```

La ejecución de la primera copia completa para todas las máquinas será costosa, ya que se copiarán varios gigas de datos. No obstante, las sucesivas copias apenas contendrán cambios, realizándose en unos pocos minutos. Por esta razón se realizarán las copias de seguridad todas las noches, ya que no serán apenas costosas ni llevarán mucho tiempo.

4.3.3. *Piojito*, servicios externos

Piojito será la máquina virtual más representativa del sistema, ya que ofrecerá los servicios más utilizados por los usuarios, que se utilizarán tanto fuera como dentro de la Universidad.

Los servicios que ofrecerá son los siguientes:

- **Web**.
- **Correo**.
- **DNS**.
- **Mysql**.
- **Intranet**.

Piojito como máquina virtual independiente

Será la máquina con más accesos externos y exposiciones a potenciales ataques, y se constituirá como una máquina aislada porque la seguridad del sistema estará adaptada a la función que ejerce; si se ejecutasen más servicios, la seguridad del sistema se reduciría drásticamente. Las medidas de seguridad más importantes serán las siguientes: acceso únicamente a los directorios de correo y web de cada cuenta, descartando los directorios *home* y *backup* que contienen datos más críticos, y ausencia de acceso por SSH, impidiendo la ejecución de comandos en la máquina, solo se ejecutarán los servicios listados anteriormente.

Cuentas de usuario en *Piojito*

En *Piojito*, es necesario instalar la librería *libnss-ldap* para que el sistema obtenga la información de cada cuenta de usuario del directorio Ldap. Esta información servirá principalmente para saber a quien pertenecen los permisos de cada fichero o directorio, así como las rutas a los directorios *home* de cada usuario.

Los directorios que importará de *Lucas* de cada cuenta de usuario son: */mnt/mail* y */mnt/web*, como se ha comentado. Los permisos de montado serán: *noexec* (impide la ejecución de ficheros) y *nodev* (impide el uso de ficheros como si fuese un dispositivo). La razón es impedir que cualquier usuario pueda ejecutar desde su cuenta cualquier tipo de comando a través de los servicios que utilizan los directorios importados.

Los directorios en *Piojito* donde se montarán */mnt/mail* y */mnt/web* de *lucas*, serán

/mnt/home y */mnt/web* respectivamente. La razón de que se monte el directorio de correo (*/mnt/mail*) como si se tratase del *home*, es para que el servidor de correo guarde los mensajes en dicho directorio, accediendo a él como si del directorio *home* se tratase. Por su parte, el servidor web accederá por defecto al *home* de usuario para buscar el directorio *public_html* o *private_html*; se crearán enlaces simbólicos *public_html* y *private_html* en cada directorio de correo (montado como */mnt/home* en *piojito*), que apunten a los respectivos directorios en */mnt/web*. En la figura 4.11 se puede ver ilustrado el proceso.

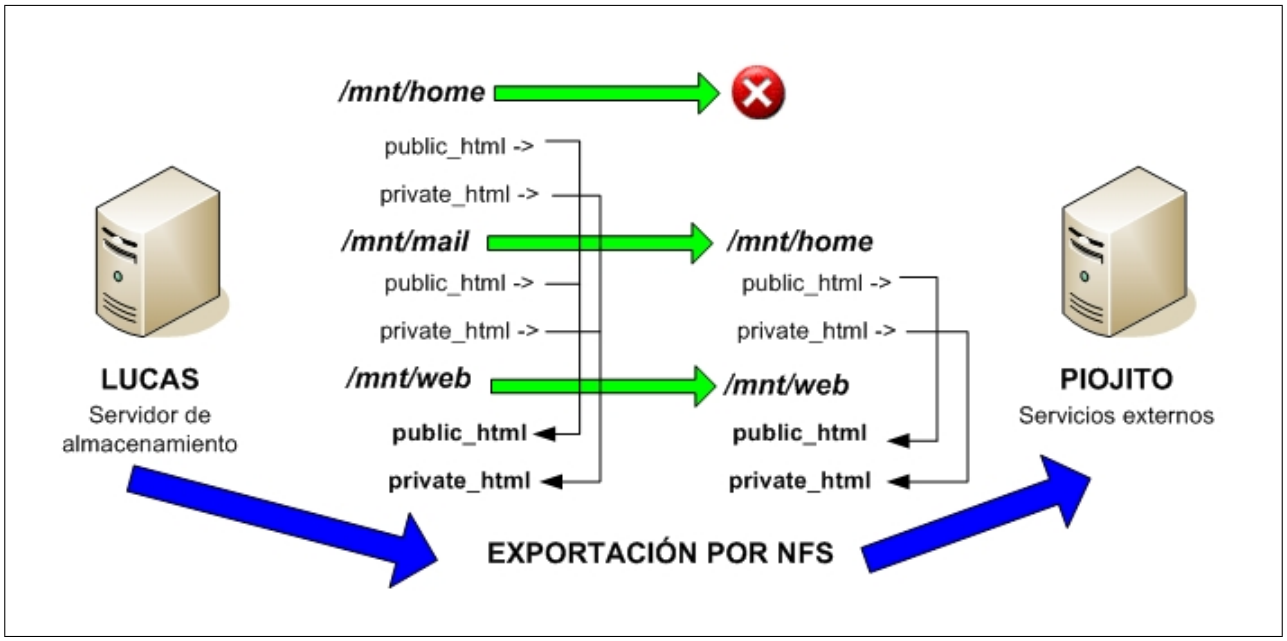


Figura 4.11: Directorios montados en *Piojito*

Servicio web en *Piojito*

Para ofrecer el servicio de *web*, *Piojito* utilizará Apache y Apache-ssl, ejecutándose en los puertos por defecto (80 para el modo normal y 443 para el modo seguro). La página principal que aparecerá al acceder al servidor web por *http* (modo normal) será la web de *ARCOS* en inglés. Esta se ejecutará desde una de las cuentas de usuario. Apache por tanto, estará configurado para servir las páginas web de las cuentas de usuario, que se montarán desde *Lucas* en el directorio */mnt/web* de *Piojito*. Tal y como aparece en la figura 4.11, Apache accederá al directorio *home* que indique la información de la cuenta en busca del subdirectorio *public_html*. El directorio de correo montado en *piojito* como *home*, contendrá en enlace simbólico que redirigirá el servidor Apache al respectivo directorio *public_html* de la cuenta, dentro de */mnt/web*.

Dado que no se pueden ejecutar programas o *scripts* desde los directorios *public_html* o *private_html* de los usuarios (según las opciones de montaje de */mnt/web*), Apache no puede ejecutar *CGIs* instalados en las cuentas de usuario. No obstante, Apache permite la ejecución de *CGIs* instalados en el sistema dentro de */var/lib/cgi-bin*. Esto se hace para evitar la ejecución de *CGIs* no controlados por los administradores; si algún usuario necesita utilizar un *CGI* propio, deberá consultar a los administradores para instalarlo dentro del directorio de *CGIs* del sistema (previa verificación de seguridad).

En Apache y Apache-ssl se instalará el módulo *libapache-auth-ldap* que permite la autenticación a través del directorio *ldap* del sistema. Por defecto, cada cuenta de usuario tendrá en su directorio *private_html* un fichero *.htaccess* configurado para autorizar el acceso al propietario de la cuenta únicamente.

Servicio de correo en *Piojito*

Una de las funciones más importantes de *Piojito* que más se van a utilizar dentro del sistema, es el servicio de correo. *Piojito* contará con Postfix como servidor *smtp* así como Courier-imap y Courier-pop como servidores Imap y Pop3 respectivamente. La principal ventaja del nuevo sistema respecto al correo es el uso de *Maildirs*, opción soportada por Postfix mediante la adicción del parámetro *home_mailbox*, e indicando el directorio que se utilizará para los *maildirs*.

Cada correo que llegue al sistema, Postfix lo encolará y redirigirá como un fichero dentro del directorio *Maildir* de la cuenta a la que vaya dirigido. Postfix estará configurado para usar el directorio *Maildir* dentro del *home* de dicho usuario, y como en *Piojito* se montará el directorio */mnt/mail* de *Lucas* como directorio */mnt/home*, Postfix accederá al directorio *Maildir* como si estuviese dentro del *home* de la cuenta. La figura 4.12 ilustra el proceso.

Piojito actuará como *relay* de correo para el segmento de *ARCOS* (163.117.148.0/24). Los correos dirigidos a cuentas de usuario de *ARCOS* los tratará *Piojito*, el resto los enviará a *smtp.uc3m.es* para enviarlos al exterior. Los correos que vengan de fuera de la Universidad pasarán a través de *smtp.uc3m.es* quien los enviará a *Piojito*, que está configurado en el *DNS* como servidor de correo (registro *MX* del dominio *ARCOS*).

En cuanto a la seguridad del correo, tanto Pop3 como Imap, funcionarán también en modo cifrado. Además se utilizará un filtro mediante Amavis, el cual utilizará el antivirus Clamav y el *antispam* Spamassassin para descartar correos con *spam* o contenido maligno. Por último, se utilizará el gestor de correo web Squirrelmail dentro de la *intranet* de *Piojito*, que se muestra a través de Apache-ssl, por tanto la comunicación será cifrada.

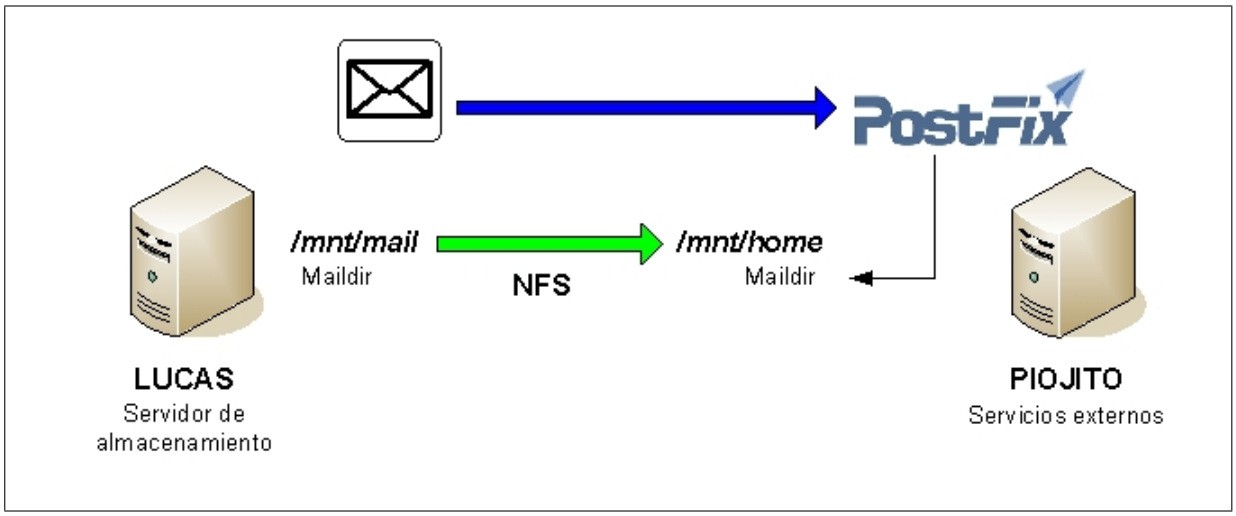


Figura 4.12: Maildirs en *piojito*

Servicio DNS en *Piojito*

Piojito también actuará como servidor DNS primario del sistema. BIND9 será el software a utilizar como servidor de DNS. Se definirá un rango de direcciones IP para realizar *dhcp*, que irán desde la 220, a la 239 dentro del rango de ARCOS (163.117.148.0/24). Además se crearán una serie de alias para facilitar la búsqueda de la máquina que tenga el servicio necesitado, siendo la lista de alias la siguiente:

```
ldap            IN CNAME piolin
www             IN CNAME piojito
ns              IN CNAME piojito
mailhost        IN CNAME piojito
mail            IN CNAME piojito
correo          IN CNAME piojito
pop3            IN CNAME piojito
pop             IN CNAME piojito
imap            IN CNAME piojito
smtp            IN CNAME piojito
dhcp            IN CNAME piojito
www2            IN CNAME piojito
www-es          IN CNAME piojito
www-en          IN CNAME piojito
intranet        IN CNAME piojito
nfs             IN CNAME lucas
smb             IN CNAME lucas
reloj           IN CNAME lucas
```

```
clock           IN CNAME lucas
disk            IN CNAME lucas
disco           IN CNAME lucas
ssh             IN CNAME caponata
wwws            IN CNAME caponata
svn             IN CNAME caponata
bugzilla        IN CNAME caponata
backup          IN CNAME boyerito
```

Servicio de bases de datos en *Piojito*

Otra de las funciones que tendrá *Piojito*, será la de servidor de bases de datos. Se utilizará MySQL como gestor de base de datos, configurado para atender únicamente peticiones locales (por el interfaz *loopback*). Las principales herramientas que utilizarán MySQL serán: la página web de ARCOS en español, la herramienta Moodle existente para el aula virtual, y la herramienta Wordpress (utilizada como noticiario de ARCOS).

Servicio de *intranet* en *Piojito*

Por último, *Piojito* contendrá la *intranet* de ARCOS, un espacio web en modo seguro (*https*), preparado para ofrecer distintos servicios al personal de ARCOS. La dirección de dicha *intranet* será: *https://arcos.inf.uc3m.es*, existiendo enlaces tanto en la web en español como en la web en inglés de ARCOS. El acceso a la *intranet* será bajo autenticación, cualquier cuenta de ARCOS tendrá acceso a la misma: Apache-ssl autenticará al usuario a través del directorio Ldap. La *intranet* tendrá dos partes claramente diferenciadas: un espacio para los usuarios normales y otro espacio para los administradores.

El espacio para los usuarios normales estará a su vez dividido en tres secciones: utilidades, documentación y la sección de entrada. Los servicios que ofrecerá cada una se listan a continuación:

- **Principal**: contendrá el servicio de correo web (Squirrelmail), un cliente SSH vía web, un enlace de ayuda para usuarios sobre las funcionalidades del sistema y acceso a las listas de correo vía web.
- **Utilidades**: utilidades diversas como una calculadora, calendarios o enlaces a buscadores, software, noticias o *blogs* interesantes.
- **Documentación**: documentación útil para el personal de ARCOS sobre docencia, investigación, etc.

El espacio para los administradores estará también dividido en tres secciones: monitorización, documentación y sección de entrada. Los servicios de cada apartado se listan a continuación:

- **Principal**: contendrá el acceso web a las distintas herramientas de configuración del sistema como: Phpmyadmin para administrar el gestor de bases de datos MySQL; Phpldapadmin, un interfaz de administración del directorio Ldap vía web; el interfaz de gestión de las listas de correo (Mailman); y por último, el interfaz de gestión vía web de la página de *ARCOS* en español.

- **Documentación**: contendrá un enlace al *wiki* de los administradores, donde se irán apuntando tareas pendientes y tareas realizadas, así como diversa información sobre la administración. También habrá un listado de las máquinas del sistema y de los servicios principales que ejecuta cada una. En cada uno de esos servicios, existirá un enlace a los ficheros de configuración, para tener un acceso rápido a dicha información en caso de necesidad.

- **Monitorización**: contendrá un apartado de servicios, donde se instalarán programas de monitorización vía web para: el almacenamiento, el servicio de resolución de nombres (DNS), el servicio de correo, el servicio web y el servicio SSH. Existirá otro apartado para las máquinas con programas de monitorización normal y pro-activa para cada uno de los servidores, así como monitorización de la red para todos los ordenadores de ARCOS.

4.3.4. *Caponata*, terminal remoto

Caponata será la máquina virtual dedicada a servir de terminal remoto SSH para el personal de ARCOS. Permitirá el acceso mediante SSH a los directorios *home*, *mail*, *web* y *BACKUP* de cada cuenta de usuario, previa autenticación. Mostrará un terminal *bash* dentro del propio servidor y permitirá la transferencia de archivos por *sftp*.

Caponata como máquina virtual independiente

La razón principal por la cual existirá como máquina aislada es el impedir que los usuarios que accedan a la misma, provoquen cualquier tipo de incidencia en otros servicios del sistema. Al ser una máquina dedicada exclusivamente a servir de terminal remoto, el único servicio que cualquier usuario, de manera intencionada o no, podría vulnerar, es el servicio de *SSH*. Además la seguridad de *Caponata* estará adaptada a dicho servicio, y tiene en cuenta los riesgos que conlleva, que se analizarán en esta sección.

Acceso a las cuentas de usuario en *Caponata*

Los directorios que montará de *Lucas* (el servidor de almacenamiento y exportación por NFS), serán */mnt/mail*, */mnt/web*, */mnt/home* y */mnt/backup*. Los permisos de exportación desde *Lucas* serán *root_squash*, ya que no se requerirá en *Caponata* que el usuario *root* tenga acceso a cualquier cuenta. Para que el sistema conozca la información

de los usuarios y permita autenticarlos dentro del sistema, estarán instaladas las librerías *libnss-ldap* y *libpam-ldap*. *Caponata* será la primera máquina del sistema que tendrá instalada la librería *libpam-ldap*, configurando el servicio SSH para que autentique a los usuarios a través del directorio Ldap, permitiéndoles el acceso al sistema.

Medidas de seguridad en *Caponata*

Caponata deja que los usuarios tengan acceso a sus datos, ya que están montados los directorios comentados anteriormente desde *Lucas*. Existe el riesgo de que los usuarios quieran obtener privilegios de manera ilícita para acceder a la información de otras cuentas. *Caponata* contará con monitorización a través de la librería *snoopy* de todas las tareas ejecutadas por cada usuario, desde que accede al sistema, advirtiendo al inicio de cada sesión sobre ello.

Por último se utilizarán medidas para evitar la sobrecarga de la máquina por cualquier usuario, impidiendo que el resto puedan ejecutar sus tareas. Para ello se utilizará el fichero */etc/security/limits.conf*, donde se configurará el tiempo máximo de ejecución de un proceso lanzado por un usuario, el número máximo de procesos a lanzar y la prioridad máxima con la que un usuario puede lanzar un proceso.

4.4. Resumen

El esquema general del nuevo sistema queda reflejado en la figura 4.13. Como se puede observar en el gráfico, aparecen las tres máquinas físicas que contendrá el sistema (*Donald*, *Daisy* y *Boyerito*), así como las cuatro máquinas virtuales (*Lucas*, *Piojito*, *Piolin* y *Caponata*).

Como se ha indicado anteriormente, las máquinas físicas *Donald* y *Daisy* están diseñadas para lo siguiente:

- Almacenar los datos de ARCOS (cuentas de usuario, copias de seguridad del sistema, etc) en volúmenes lógicos, creados a partir de dispositivos RAID. Los volúmenes lógicos se llamarán: *usuarios* (almacenará los datos de cada cuenta de usuario), *backup_usuarios* (almacenará los datos que cada usuario considere más críticos), y *sistema* (almacenará las copias de seguridad de las máquinas y servicios del sistema).

- Ejecutar las máquinas virtuales mediante Xen y servir como máquina anfitriona.

La diferencia entre *Donald* y *Daisy* es que *Donald* será la máquina encargada de realizar las tareas anteriores y *Daisy* será la máquina de respaldo en caso de que *Donald* falle. Los datos de ARCOS que se almacenarán en los volúmenes lógicos, se utilizarán desde

Donald, y se realizará una sincronización diaria en *Daisy*.

Por último,respecto a las máquinas físicas, *Boyerito* será la encargada de almacenar un *backup* de la información de ARCOS que contienen los volúmenes lógicos.

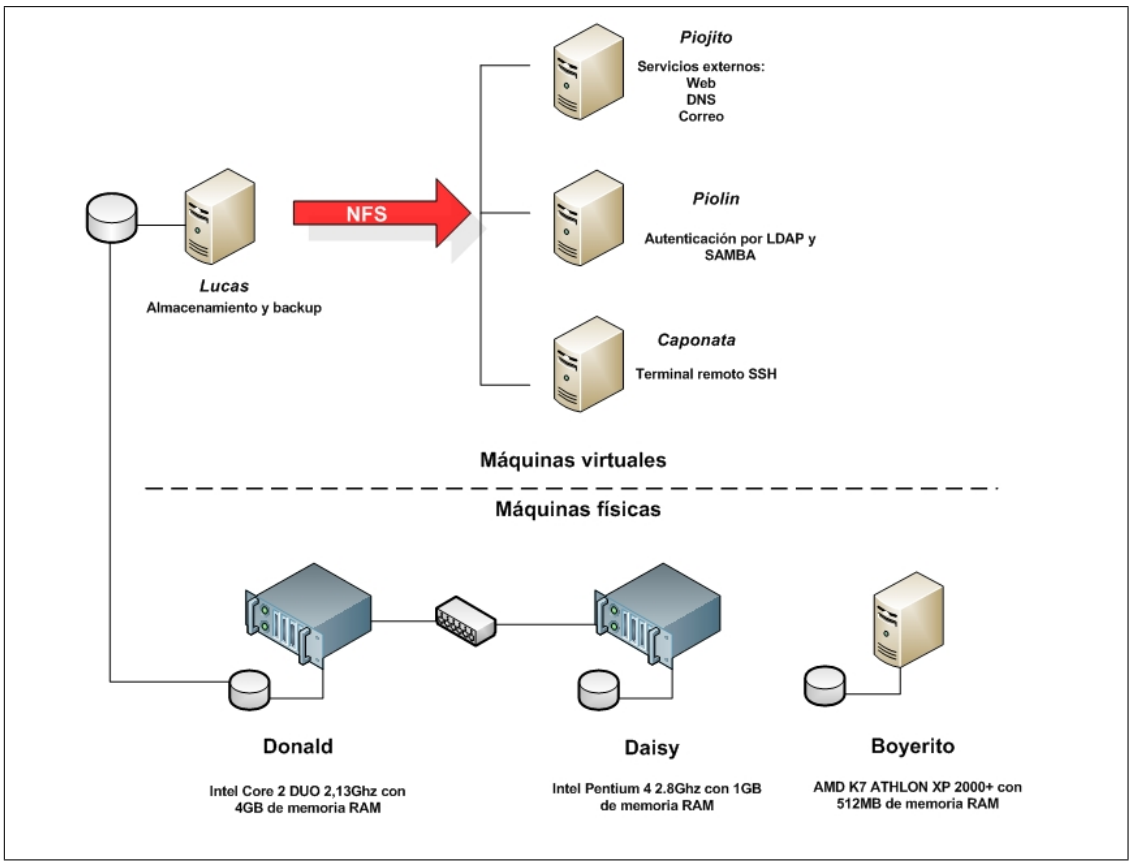


Figura 4.13: Esquema general del nuevo sistema de servidores

En cuanto a las máquinas virtuales, *Lucas* tendrá las siguientes funciones:

- Servir por NFS al resto de máquinas la información contenida en los volúmenes lógicos de la máquina anfitriona (*Donald* si el sistema se ejecuta de modo normal o *Daisy* si el sistema funciona en modo respaldo).
- Realizar las copias de seguridad de las máquinas y servicios del sistema, almacenándolas en el volumen lógico *sistema*.

Piojito se encargará de los servicios externos, es decir:

- Servidor web.

- Servidor de correo.
- Servidor DNS.

Piolin por su parte se encargará de la autenticación por Ldap y Samba y por último *Caponata* servirá de terminal remoto SSH.

Por último, el orden de inicio de las máquinas virtuales será el siguiente:

- ***Lucas***: es la primera en iniciarse dado que necesita exportar por NFS los datos de los volúmenes lógicos a las demás máquinas virtuales:.
- ***Piolin***: es la segunda porque tanto *Piojito* como *Caponata* utilizarán la autenticación por Ldap para los servicios que prestan.
- ***Piojito***: es la tercera porque necesita que *Lucas* y *Piolin* están iniciadas para utilizar los datos de los usuarios y la autenticación Ldap, y sus servicios son también críticos.
- ***Caponata***: es la cuarta porque el servicio que ofrece es el menos relevante respecto a las otras máquinas virtuales.

Capítulo 5

Implantación

5.1. Fase I: instalación y configuración de máquinas

El primer paso para la implantación del sistema es instalar los sistemas operativos en las máquinas físicas, para luego dar paso a la creación de las máquinas virtuales. Se instalará el sistema operativo también en éstas máquinas y por último se irán configurando los servicios.

5.1.1. Instalación de los sistemas operativos en las máquinas físicas

Las máquinas físicas son: *Donald*, *Daisy* y *Boyerito*. *Donald* y *Daisy* han de estar configuradas como máquinas anfitrionas de Xen, por lo tanto, necesitarán cargar un *kernel* adaptado para realizar paravirtualización con Xen versión 3. El documento donde se detalla la instalación de *Donald* aparece como anexo en el presente documento (ver página 220.), al igual que el documento donde se detalla la instalación de *Daisy* (ver página 232.). Por último, la instalación de *Boyerito* aparece como anexo en la página 241.

5.1.2. Creación de máquinas virtuales

Una vez instalada *Donald* y funcionando como *Dom0* para Xen, se deben crear los directorios que alojarán las máquinas virtuales y los respectivos ficheros de imagen que contendrán el sistema operativo de cada máquina virtual.

Se utilizará el comando *xen-create-image* para la creación de dichas imágenes con el sistema operativo instalado y con algunas configuraciones realizadas sobre él. Las máquinas a crear y los comandos ejecutados se listan a continuación:

- ***Piojito***: ésta máquina virtual tendrá 10 *Gigas* de capacidad dado que será la que más espacio necesite, al tener que instalarse en ella más programas que en las demás.

El comando a ejecutar será:

```
xen-create-image --fs=ext3 --image=full --kernel=/boot/vmlinuz-2.6.18.8-xen \
--memory=1490 --passwd --size=10G --swap=512M --ide --dist=etch \
--debootstrap --ip=163.117.148.240 --netmask=255.255.255.0 \
--gateway=163.117.148.2 --dir=/maquinas --hostname=piojito
```

- ***Piolin***: ésta máquina virtual tendrá 5 *Gigas* de capacidad, el comando a ejecutar será:

```
xen-create-image --fs=ext3 --image=full --kernel=/boot/vmlinuz-2.6.18.8-xen \
--memory=512 --passwd --size=5G --swap=512M --ide --dist=etch --debootstrap \
--ip=163.117.148.241 --netmask=255.255.255.0 --gateway=163.117.148.2 \
--dir=/maquinas --hostname=piolin
```

- ***Lucas***: ésta máquina virtual tendrá 4 *Gigas* de capacidad, necesitará de al menos 1 *Giga* de *RAM*, porque realizará las copias de seguridad mediante *rsync* y necesitará al menos esa cantidad de memoria para alojar la estructura de directorios. El comando a ejecutar será:

```
xen-create-image --fs=ext3 --image=full --kernel=/boot/vmlinuz-2.6.18.8-xen \
--memory=1024 --passwd --size=4G --swap=512M --dist=etch --debootstrap \
--ip=163.117.148.244 --netmask=255.255.255.0 --gateway=163.117.148.2 \
--dir=/maquinas --hostname=lucas
```

- ***Caponata***: ésta máquina virtual tendrá 5 *Gigas* de capacidad y 512 *Mb* de memoria. El comando a ejecutar será:

```
xen-create-image --fs=ext3 --image=full --kernel=/boot/vmlinuz-2.6.18.8-xen \
--memory=512 --passwd --size=5G --swap=512M --ide --dist=etch --debootstrap \
--ip=163.117.148.245 --netmask=255.255.255.0 --gateway=163.117.148.2 \
--dir=/maquinas --hostname=caponata
```

Para que las máquinas virtuales se inicien al ejecutarse *Donald*, se crean los siguientes enlaces simbólicos en */etc/xen/auto*:

```
ln -s /maquinas/lucas/lucas.cfg /etc/xen/auto/10-lucas
ln -s /maquinas/piolin/piolin.cfg /etc/xen/auto/20-piolin
ln -s /maquinas/piojito/piojito.cfg /etc/xen/auto/30-piojito
ln -s /maquinas/caponata/caponata.cfg /etc/xen/auto/40-caponata
```

Los nombres de dichos enlaces siguen el orden en el cual han de iniciarse las máquinas. Por último, para comprobar que funcionan correctamente, se reinicia *Donald* y se comprueba que todas las máquinas virtuales están funcionando:

```
# xm list
Name                                        ID Mem(MiB) VCPUs State   Time(s)
Domain-0                                     0      512     2 r-----  29088.4
caponata                                    14      512     1 -b----  18790.5
lucas                                       11     1024     1 -b----  43300.1
piojito                                     13     1490     1 -b---- 125106.7
piolin                                      12      512     1 -b----   2323.5
```

5.2. Fase II: instalación y configuración de los servicios de autenticación y almacenamiento

El siguiente paso para la implantación del sistema es la configuración de los servicios que integra cada máquina.

5.2.1. Instalación y configuración de los servicios de autenticación Ldap

En primer lugar será necesario instalar el servidor Ldap en *Piolin* y dejarlo activo, a continuación se instalará el paquete de *scripts smbldap-tools*, se configurará y se crearán las entradas necesarias en el directorio Ldap. Para que los sistemas Linux puedan acceder a la información de las cuentas de usuario contenidas en el directorio Ldap, se instalará la librería *libnss-ldap* en todas las máquinas virtuales; si además se desea autenticar usuarios en el propio sistema Linux (en el caso de Caponata), se instalará la librería *libpam-ldap*. Por último, se instalará y configurará el servidor Samba (en *Piolin* y en *Lucas*) para autenticar a través del directorio Ldap. Los siguientes apartados describirán cada uno de estos pasos.

Instalación del servidor Ldap

En *Piolin*, se instalan los paquetes necesarios y todas las dependencias que necesiten:

```
#apt-get install slapd ldap-utils samba-doc
```

Se genera una contraseña para el administrador del Ldap: (ejemplo con contraseña 'pepito')

```
# slappasswd
New password:
Re-enter new password:
{SSHA}pCmTeZCli/V5E5qdsJXG5PiDlEyJFb1r
```

El paquete *samba-doc* es necesario porque incluye el *schema* de Samba para Ldap. Es necesario copiar este fichero a la carpeta que contiene los ficheros *schema* del Ldap:

```
# cd /usr/share/doc/samba-doc/examples/LDAP
# gunzip samba.schema.gz
# cp samba.schema /etc/ldap/schema
```

Se crea el fichero de configuración */etc/ldap/slapd.conf* para el servidor Ldap (ver página 248.). Por orden, en el fichero aparecen los ficheros *schema* a utilizar (campos *include*), el tipo de acceso a los distintos atributos de cada entrada en el Ldap: los atributos que son contraseñas solo pueden ser reescritos por el propio usuario, o accedidos mediante contraseña de dicho usuario (*by self write, by * auth*), y existe un usuario llamado 'consultas' para no realizarlas de forma anónima, sino mediante dicho usuario; se define el sufijo del arbol Ldap (*suffix "dc=arcos,dc=inf.uc3m.es"*) y la entrada del administrador (*rootdn "cn=admin,dc=arcos,dc=inf.uc3m.es"*), así como la contraseña del mismo (*rootpw *******); y por último aparecen los atributos escogidos para crear índices.

Por último, antes de arrancar el servicio, es necesario que la carpeta */var/lib/ldap* contenga el fichero *DB_CONFIG*. Para ello, se copia el fichero de ejemplo que se instala con el paquete a dicho directorio:

```
# cp /usr/share/slapd/DB_CONFIG /var/lib/ldap/.
```

Finalmente se inicia el servidor Ldap:

```
# /etc/init.d/slapd start
```

Instalación de las herramientas *smbldap-tools* [4]

En *Piolin*, se instala el paquete *smbldap-tools* y todas las dependencias que necesite:

```
# apt-get install smbldap-tools samba-common
```

Se configura el fichero */etc/smbldap-tools/smbldap.conf* (ver página 249.). Para el atributo *SID*, se ejecuta el comando de Samba *net* (razón por la que se instaló el paquete *samba-common*) de la siguiente forma:

```
# net getlocalsid
SID for domain PIOLIN is: S-1-5-21-3492619381-3135118558-3133272105
```

El *SID* que aparece es el que hay que introducir el la línea correspondiente del fichero *smbldap.conf*. Hay que definir el nombre del dominio en el atributo *sambaDomain* (será ARCOS), definir el *host* donde se realizarán las peticiones Ldap (en este caso será *localhost* y utilizando el puerto por defecto: 389), el nombre dentro del árbol, con los sufijos añadidos, de los atributos que definen las personas, los grupos y las computadoras; el tipo de *hash* para la encriptación de las contraseñas; y por último, diversa información sobre las cuentas para Unix y para Windows.

Se configura el fichero */etc/smbldap-tools/smbldap_bind.conf* (ver página 254). Este contiene la dirección de los *host* primario y esclavo donde realizar las consultas Ldap (ambos serán *localhost*), y la contraseña de administrador para autenticarse contra el servidor Ldap.

Por último se ejecuta el comando *smbldap-populate* que generará la estructura necesaria de usuarios y grupos para poder ser un controlador de dominio de Windows:

```
# smbldap-populate
Using builtin directory structure
adding new entry: dc=arcos,dc=inf.uc3m.es
adding new entry: ou=Users,dc=arcos,dc=inf.uc3m.es
adding new entry: ou=Groups,dc=arcos,dc=inf.uc3m.es
adding new entry: ou=Computers,dc=arcos,dc=inf.uc3m.es
adding new entry: ou=Idmap,dc=arcos,dc=inf.uc3m.es
adding new entry: cn=NextFreeUnixId,dc=arcos,dc=inf.uc3m.es
adding new entry: uid=Administrator,ou=Users,dc=arcos,dc=inf.uc3m.es
adding new entry: uid=nobody,ou=Users,dc=arcos,dc=inf.uc3m.es
adding new entry: cn=Domain Admins,ou=Groups,dc=arcos,dc=inf.uc3m.es
adding new entry: cn=Domain Users,ou=Groups,dc=arcos,dc=inf.uc3m.es
adding new entry: cn=Domain Guests,ou=Groups,dc=arcos,dc=inf.uc3m.es
adding new entry: cn=Print Operators,ou=Groups,dc=arcos,dc=inf.uc3m.es
adding new entry: cn=Backup Operators,ou=Groups,dc=arcos,dc=inf.uc3m.es
adding new entry: cn=Replicator,ou=Groups,dc=arcos,dc=inf.uc3m.es
adding new entry: cn=Domain Computers,ou=Groups,dc=arcos,dc=inf.uc3m.es
```

Se define una contraseña para el usuario *Administrator*:

```
# smbldap-passwd administrator
Changing UNIX and samba passwords for administrator
New password:
Retype new password:
```

Con el comando *smbldap-usershow* se lista la información de un usuario en concreto. Para verificar la correcta ejecución de las herramientas *smbldap-tools* y del servidor Ldap se realiza una consulta:

```
# smbldap-usershow Administrator
dn: uid=administrator,ou=People,dc=arcos,dc=inf.uc3m.es
objectClass: top,person,organizationalPerson,inetOrgPerson,
sambaSamAccount,posixAccount,shadowAccount
gidNumber: 0
uidNumber: 0
homeDirectory: /home/root
sambaLogonTime: 0
sambaLogoffTime: 2147483647
sambaKickoffTime: 2147483647
sambaHomePath: \\%L\homes
sambaHomeDrive: Z:
sambaProfilePath: \\%L\profiles
sambaPrimaryGroupSID: S-1-5-21-3492619381-3135118558-3133272105-512
sambaSID: S-1-5-21-3492619381-3135118558-3133272105-500
sambaPasswordHistory: 00000000000000000000000000000000
sambaPwdCanChange: 1152294458
uid: administrador
gecos: Netbios Domain Administrator,,,,
cn: Netbios Domain Administrator
sn: Administrator
givenName: Netbios Domain
loginShell: /bin/false
sambaLMPassword: xxxxxxxxxxxxxxxxxxxxxxxx
sambaAcctFlags: [U]
sambaNTPassword: xxxxxxxxxxxxxxxxxxxxxxxx
sambaPwdLastSet: 1180716992
sambaPwdMustChange: 1184604992
userPassword: {MD5}xxxxxxxxxxxxxxxxxxxxxx
```

Se obtiene un resultado correcto. Se puede comprobar como el *uidNumber* y *gidNumber* del usuario *Administrator* son cero, lo que equivale a decir que es el usuario *root* para el sistema. Por seguridad se deniega el acceso a una *shell* (*loginShell: /bin/false*).

Instalación de la librería *libnss-ldap*

Para que todas las máquinas virtuales tengan acceso a la información de cada cuenta de usuario, es necesaria la librería *libnss-ldap*. En todas las máquinas virtuales (*Piolin*, *Piojito*, *Lucas* y *Caponata*) se instalará mediante:

```
# apt-get install libnss-ldap
```

Una vez instalado el paquete se modificará el fichero */etc/nsswitch.conf* (ver página 259.) para que el sistema Linux busque la información de los usuarios en los ficheros */etc/passwd,*

/etc/shadow, /etc/group, y en el directorio Ldap.

También es necesario indicar los parámetros de búsqueda dentro del directorio Ldap. Éstos parámetros se definen en el fichero */etc/libnss-ldap.conf*, ver página 260.

Para comprobar el correcto funcionamiento de la librería se utiliza el comando *finger*:

```
# finger administrator
Login: administrator                    Name: Netbios Domain Administrator
Directory: /home/root                   Shell: /bin/false
Last login Sun Sep  9 16:51 (CEST) on pts/2 from piolin.arcos.inf.uc3m.es
No mail.
No Plan.
```

Instalación de la librería *libpam-ldap*

La máquina virtual *Caponata* permitirá acceso por SSH a la misma, por lo tanto necesitará autenticar a los usuarios a través del directorio Ldap. En el caso de la máquina virtual *Piojito*, los servicios de correo (Imap y Pop3) necesitarán también autenticar a cada usuario del mismo modo. Para permitir este tipo de autenticación se instalará en ambas máquinas la librería *libpam-ldap*:

```
# apt-get install libpam-ldap
```

El fichero de configuración donde leerá los parámetros de acceso al directorio Ldap será */etc/ pam_ldap.conf* (ver página 260.) que contiene la misma información que el fichero */etc/nsswitch.conf*.

La librería *pam* de Linux es la que utilizan varios de los servicios del sistema que necesitan autenticación. Los ficheros de configuración de dichos servicios están en el directorio */etc/pam.d*. En el caso de *Caponata* y *Piojito* se configurarán los ficheros */etc/pam.d/common-account* (ver página 261.), */etc/pam.d/common-auth* (ver página 261.) y */etc/pam.d/common-password* (ver página 261.) que contienen la configuración general usada para el acceso al sistema. En cada una de las máquinas, cada servicio que utiliza *pam*, incluirá estos tres ficheros. En el caso de *Piojito*, se desea denegar el acceso por SSH, no incluyendo en el fichero */etc/pam.d/ssh* los tres ficheros comentados anteriormente (ver página 262.).

Instalación del servidor Samba de autenticación y de servidor de disco

Tanto en *Piolin* como en *Lucas* y en *Caponata*, se instala el paquete *samba*:

```
# apt-get install samba
```

Se introduce la contraseña del servidor Ldap donde Samba realizará las peticiones: (ejemplo con contraseña 'pepito')

```
# smbpasswd -w 'pepito'
Setting stored password for "cn=admin,dc=arcos,dc=inf.uc3m.es" in secrets.tdb
```

Para *Piolin*, se configura el fichero */etc/samba/smb.conf* como aparece en el anexo 4 del presente documento (ver página 254.). El servidor Samba de *Piolin*, contemplará los siguientes puntos:

- Funcionará como controlador primario de dominio; para realizar esto se configuran los atributos: *domain logons*, *os level*, *preferred master*, *domain master* y *local master* con los valores que aparecen en el anexo.
- La pasarela de autenticación que utilizará Samba, será el servidor Ldap, (atributo *passdb backend*).
- Se utilizarán los *scripts* que integra el paquete *smbldap-tools* para realizar las tareas de administración de usuarios y grupos.
- El nombre del dominio Samba será Arcos, y el nombre del controlador primario de dominio (servidor Samba de *Piolin*) será *Piolin*.

Para *Lucas*, se configura también el fichero */etc/samba/smb.conf* como aparece en el anexo 4 del presente documento (ver página 256.). El servidor Samba de *Lucas*, contemplará los siguientes puntos:

- Funcionará como otra máquina de autenticación (no siendo controlador primario de dominio); esto de define mediante los atributos: *domain logons*, *os level*, *preferred master*, *domain master* y *local master* con los valores que aparecen en el anexo.
- La pasarela de autenticación que utilizará Samba, será el servidor Ldap, (atributo *passdb backend*).
- El nombre del dominio Samba a utilizar será ARCOS, y el nombre del servidor Samba de *Lucas* será *disco*.
- Dado que el servidor Samba de *Lucas* actuará de servidor de disco, ofrecerá los siguientes recursos:
  - **Recurso *homes***: este recurso será el que de acceso al directorio *home* de cada cuenta de usuario. No se define la ruta dado que Samba ha de obtenerla del sistema.

- **Recurso *netlogon***: este recurso será el que indique la ruta donde encontrar el fichero *netlogon* de cada cuenta de usuario. Por defecto, se utilizará el mismo fichero *netlogon* para todos, contenido en el directorio */mnt/usuarios/home/netlogon/* y denominado *'netlogon.bat'*. El contenido de este fichero es:

  ```
  NET USE H: \\DISCO\BACKUP
  NET USE Z: \\DISCO\HOMES
  ```

  Provoca que al iniciar sesión en una máquina con Windows unido al domino Samba Arcos, aparezcan como unidades *H* y *Z* los recursos *Backup* y *Homes* respectivamente.
- **Recurso *profiles***: por defecto no se utilizará, ya que no se desean perfiles móviles en el sistema.
- **Recurso *backup*** : este recurso dará acceso a los directorios *BACKUP* de cada cuenta de usuario. La ruta de los mismos será */mnt/backup_usuarios/[nombre de usuario]*, y se configura mediante el atributo *path*.

Para Caponata, también se configura el fichero */etc/samba/smb.conf* como aparece en el anexo 4 del presente documento (ver página 258.). El servidor Samba de Caponata, solo se ejecuta para que el comando *smbpasswd* funcione correctamente. Este comando se utiliza para cambiar la contraseña de los usuarios que contenga el servidor Samba, que a su vez son los usuarios del servidor Ldap. El servidor Samba ejecutado en Caponata, proporciona al comando *smbpasswd* el modo de comunicarse con el directorio Ldap, para actualizar la contraseña del usuario.

Para finalizar, se iniciará el servidor Samba en *Piolin*, en *Lucas* y en *Caponata*:

```
# /etc/init.d/samba start
Starting Samba daemons: nmbd smbd.
```

5.2.2. Instalación y configuración de los servicios de disco distribuido

Todas las máquinas virtuales necesitan acceso a los volúmenes lógicos que contienen *Donald* y *Daisy*, donde residirá toda la información de las cuentas de usuario. *Lucas* es la máquina virtual que tiene acceso directo a éstos volúmenes lógicos y desde la cual se exporta por NFS al resto de máquinas. En las siguientes secciones se describirá la implementación de este diseño.

Acceso a los volúmenes lógicos en *Lucas*

Lucas tendrá acceso a los volúmenes lógicos como si de particiones de discos duros se tratasen; la correspondencia es la que aparece en la figura 5.1. Por tanto, en *Lucas*,

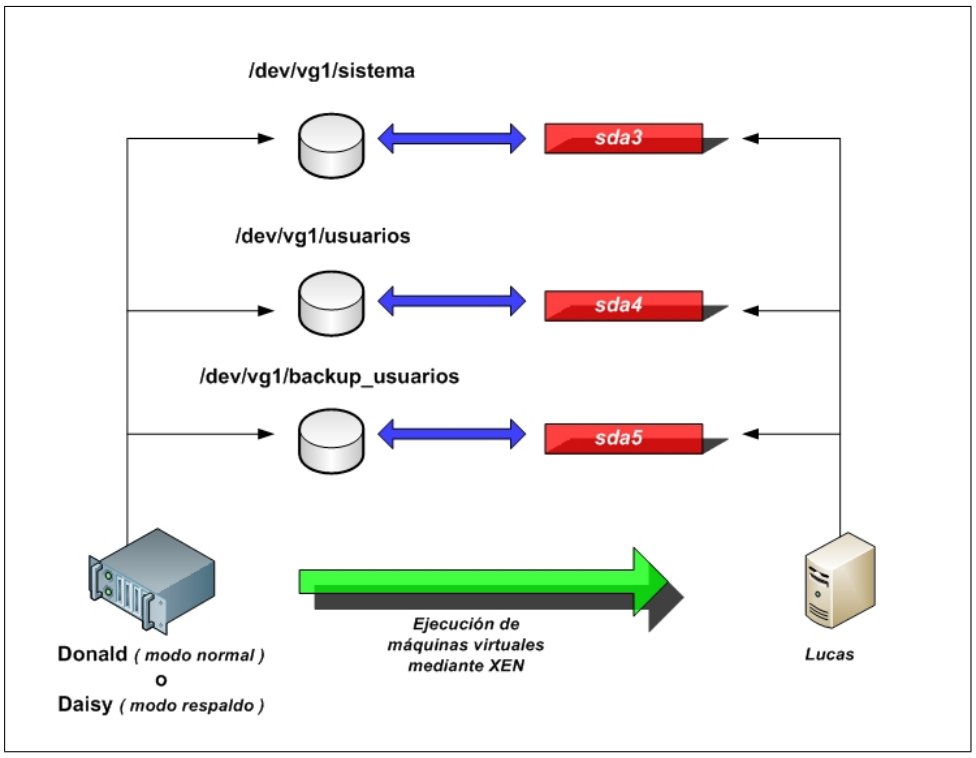


Figura 5.1: Correspondencia de los volúmenes lógicos con las particiones de *Lucas*

se tendrán las particiones *sda3* (sistema), *sda4* (usuarios) y *sda5* (backup_usuarios). Se formatean con el sistema de ficheros *ext3*:

```
# mkfs.ext3 /dev/sda3
# mkfs.ext3 /dev/sda4
# mkfs.ext3 /dev/sda5
```

Después se crean los siguientes directorios, donde irán montadas dichas particiones:

```
# mkdir -p /mnt/usuarios
# mkdir -p /mnt/backup_usuarios
# mkdir -p /mnt/sistema
```

Se introducen en el fichero */etc/fstab* las líneas necesarias para que se monten automáticamente las particiones *sda3*, *sda4* y *sda5* en los directorios anteriores:

```
# echo '/dev/sda3   /mnt/sistema          ext3   errors=remount-ro   0  2' >> /etc/fstab
# echo '/dev/sda4   /mnt/usuarios         ext3   errors=remount-ro   0  3' >> /etc/fstab
# echo '/dev/sda5   /mnt/backup_usuarios  ext3   errors=remount-ro   0  4' >> /etc/fstab
```

Se montan con el comando *mount* y se comprueba que están montadas:

```
# mount -a
# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1              4128448    577012   3341724  15% /
tmpfs                   524364         0    524364   0% /lib/init/rw
udev                     10240        36     10204   1% /dev
/dev/sda4            373321520     21000 373300520   1% /mnt/usuarios
/dev/sda5            373321520     21000 373300520   1% /mnt/backup_usuarios
/dev/sda3            309637120     18000 309619120   1% /mnt/sistema
```

Por último, se deben crear los directorios *home*, *mail* y *web* dentro de */mnt/usuarios*:

```
# mkdir /mnt/usuarios/home
# mkdir /mnt/usuarios/mail
# mkdir /mnt/usuarios/web
```

Exportación por NFS desde *Lucas*

Para exportar por NFS se instalará el paquete *nfs-kernel-server* en *Lucas* y todas sus dependencias:

```
# apt-get install nfs-kernel-server
```

Se configurará el fichero */etc/exports* (ver página 262.) para que cumpla los parámetros indicados en el capítulo de diseño (ver página 102.). Una vez creado el fichero exports, se procederá a iniciar el servidor NFS:

```
# /etc/init.d/nfs-kernel-server start
```

Para comprobar si el servidor NFS funciona correctamente, se utiliza el comando *rpcinfo*, donde se puede comprobar que se está ejecutando en los puertos 2049 de *tcp* y 2049 de *udp*:

```
# rpcinfo -p localhost
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  33311  status
    100024    1   tcp  46244  status
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

```
100003    3   tcp   2049  nfs
100003    4   tcp   2049  nfs
100021    1   udp  33342  nlockmgr
100021    3   udp  33342  nlockmgr
100021    4   udp  33342  nlockmgr
100021    1   tcp  54652  nlockmgr
100021    3   tcp  54652  nlockmgr
100021    4   tcp  54652  nlockmgr
100005    1   udp    993  mountd
100005    1   tcp    996  mountd
100005    2   udp    993  mountd
100005    2   tcp    996  mountd
100005    3   udp    993  mountd
100005    3   tcp    996  mountd
```

Importación de los datos de las cuentas de usuario

El resto de máquinas virtuales (*Piojito*, *Piolin* y *Caponata*), así como *Donald*, necesitan acceso a los directorios exportados por *Lucas*. Para poder importar un sistema de ficheros por NFS, cada una de las máquinas debe tener instalado el paquete *nfs-common* y *portmap*, por lo tanto se instalan estos paquetes en las máquinas comentadas anteriormente:

```
# apt-get install nfs-common portmap
# /etc/init.d/portmap start
```

Para importar por NFS los directorios, cada máquina los incorpora a su fichero */etc/fstab*. De esta forma, al iniciar la máquina virtual, se montan los directorios automáticamente. Las opciones de montaje varían en cada máquina, no obstante, todas tienen en común la opción *tcp*, que obliga al protocolo NFS a utilizar *tcp* en lugar de *udp*. Esto permite una conexión más estable, opción recomendable para éste sistema. A continuación se detalla como importa cada máquina los directorios que necesita:

- ***Piojito***: ésta máquina necesita acceso a los directorios */mnt/usuarios/mail* y */mnt/usuarios/ web* de *Lucas*. Se añaden las siguientes líneas al fichero */etc/fstab* y después se montan los directorios:

  ```
  # echo ‘‘163.117.148.244:/mnt/usuarios/mail /mnt/home   nfs \
                      defaults,noexec,nodev,nosuid,tcp  0 0’’ >> /etc/fstab
  # echo ‘‘163.117.148.244:/mnt/usuarios/web  /mnt/web    nfs  \
                      defaults,noexec,nodev,nosuid,tcp  0 0’’ >> /etc/fstab
  # mount -a
  ```

Las opciones de montaje *noexec*, *nodev* y *nosuid*, permiten montar el sistema de ficheros con las siguientes restricciones respectivamente: no se pueden ejecutar comandos situados en él, evita el tratamiento de ficheros como dispositivos y por último, no hace efectivos los permisos de tipo *set-user-identifier* o *set-group-identifier*.

- ***Piolin***: ésta máquina necesita acceso a todos los directorios que contienen datos de cuentas de usuario, es decir */mnt/usuarios/home*, */mnt/usuarios/mail*, */mnt/usuarios/web* y */mnt/backup_usuarios*. Se añaden las siguientes líneas al fichero */etc/fstab* y después se montan los directorios:

```
# echo ‘‘163.117.148.244:/mnt/usuarios/home   /mnt/home    nfs \
                    defaults,tcp   0 0’’ >> /etc/fstab
# echo ‘‘163.117.148.244:/mnt/usuarios/mail   /mnt/mail    nfs \
                    defaults,tcp   0 0’’ >> /etc/fstab
# echo ‘‘163.117.148.244:/mnt/usuarios/web    /mnt/web     nfs \
                    defaults,tcp   0 0’’ >> /etc/fstab
# echo ‘‘163.117.148.244:/mnt/backup_usuarios /mnt/backup  nfs \
                    defaults,tcp   0 0’’ >> /etc/fstab
# mount -a
```

  Los directorios montados no precisan de opciones de montaje especiales.

- ***Caponata***: ésta máquina necesita acceso a los mismos directorios que *Piolin*. Se añaden las siguientes líneas al fichero */etc/fstab* y después se montan los directorios:

```
# echo ‘‘163.117.148.244:/mnt/usuarios/home   /mnt/home    nfs \
                    defaults,tcp   0 0’’ >> /etc/fstab
# echo ‘‘163.117.148.244:/mnt/usuarios/mail   /mnt/mail    nfs \
                    defaults,tcp   0 0’’ >> /etc/fstab
# echo ‘‘163.117.148.244:/mnt/usuarios/web    /mnt/web     nfs \
                    defaults,tcp   0 0’’ >> /etc/fstab
# echo ‘‘163.117.148.244:/mnt/backup_usuarios /mnt/backup  nfs \
                    defaults,tcp   0 0’’ >> /etc/fstab
# mount -a
```

  Los directorios montados no precisan de opciones de montaje especiales, no obstante, es importante destacar que desde *Lucas* se exportan con opción *root_squash* para que el usuario *root* no tenga acceso a dichos directorios.

- ***Donald***: ésta máquina física necesita acceso a los directorios principales donde *Lucas* monta los volúmenes lógicos. Se añaden las siguientes líneas al fichero */etc/fstab* y después se montan los directorios:

```
# echo ‘‘163.117.148.244:/mnt/usuarios        /mnt/usuarios        nfs \
                    ro,noauto,noexec,nodev  0 0’’ >> /etc/fstab
# echo ‘‘163.117.148.244:/mnt/backup_usuarios /mnt/backup_usuarios nfs \
                    ro,noauto,noexec,nodev  0 0’’ >> /etc/fstab
# echo ‘‘163.117.148.244:/mnt/sistema         /mnt/sistema         nfs \
                    ro,noauto,noexec,nodev  0 0’’ >> /etc/fstab
# mount -a
```

Los directorios montados solo se utilizan para realizar la sincronización de datos entre *Donald* y *Daisy*, por tanto, solo es necesario que se monten cuando tenga lugar la sincronización y con las opciones que solo permitan un acceso de lectura a los datos. Éstas opciones son: *ro*, *noauto*, *noexec* y *nodev*, que permiten montar el sistema de ficheros en modo solo lectura, de manera manual, sin poder ejecutar ficheros y evitando el tratamiento de ficheros como dispositivos.

5.3. Fase III: migración de las cuentas del antiguo sistema al nuevo sistema

El antiguo sistema disponía de tres servidores: *Aguila*, *Cuervo* y *Codorniz*. *Aguila* tiene cuentas de usuario de profesores, becarios, asignaturas y proyectos; *Cuervo* solo tiene cuentas de alumnos; y por último, *Codorniz*, tiene datos de cuentas obsoletas, dado que se dejó de utilizar el dominio y los perfiles móviles. Del antiguo sistema se necesita migrar las cuentas de *Aguila*, dado que las que existen en *Cuervo* no son importantes y las de *Codorniz* son obsoletas.

Para realizar la migración se necesitará crear los grupos de usuarios, a continuación migrar toda la información de las cuentas de usuario, y por último, crear *scripts* para facilitar ciertas tareas de administración. Los pasos para llevar a cabo la migración se describen en las siguientes secciones.

5.3.1. Creación de grupos

Los grupos que se utilizarán en el nuevo sistema son los que existían en el antiguo *Aguila*, incluyendo los identificadores (*guid*). A continuación se lista el nombre de los grupos y el identificador (*guid*) asignado:

- Docencia (1000)
- Profesores (2000)

- Becarios (3000)
- Proyectos (4000)
- Alumnos (5000)
- Otros (6000)
- Masters (7000)
- Invitados (8000)

Para crear los grupo dentro del directorio Ldap se ejecutan los siguientes comandos desde *Piolin*:

```
# smbldap-groupadd -a  -g 1000 docencia
# smbldap-groupadd -a  -g 2000 profesores
# smbldap-groupadd -a  -g 3000 becarios
# smbldap-groupadd -a  -g 4000 proyectos
# smbldap-groupadd -a  -g 5000 alumnos
# smbldap-groupadd -a  -g 6000 otros
# smbldap-groupadd -a  -g 7000 masters
# smbldap-groupadd -a  -g 8000 invitados
```

5.3.2. Migración de la información de las cuentas de usuario

En *Aguila*, la información de cada cuenta de usuario estaba almacenada en los ficheros */etc/passwd*, */etc/shadow* y */etc/group*. En el nuevo sistema, el directorio Ldap contendrá dicha información, además de otros atributos adicionales, como la información de Samba.

Se deben copiar los ficheros */etc/passwd* y */etc/shadow* de *Aguila* a *Piolin* y desde alli editarlos para eliminar los usuarios creados para el propio sistema como: *root*, *www-data*, *daemon*, etc. Una vez copiados y editados los ficheros en *Piolin*, se utiliza el *script smbldap-migrate-unix-accounts* de las herramientas Smbldap-Tools:

```
# smbldap-migrate-unix-accounts -P passwd -S shadow -a
```

Cuando el comando finaliza, todas las cuentas de usuario existentes en *Aguila* están incorporadas al directorio Ldap, incluyendo las contraseñas de cada usuario.

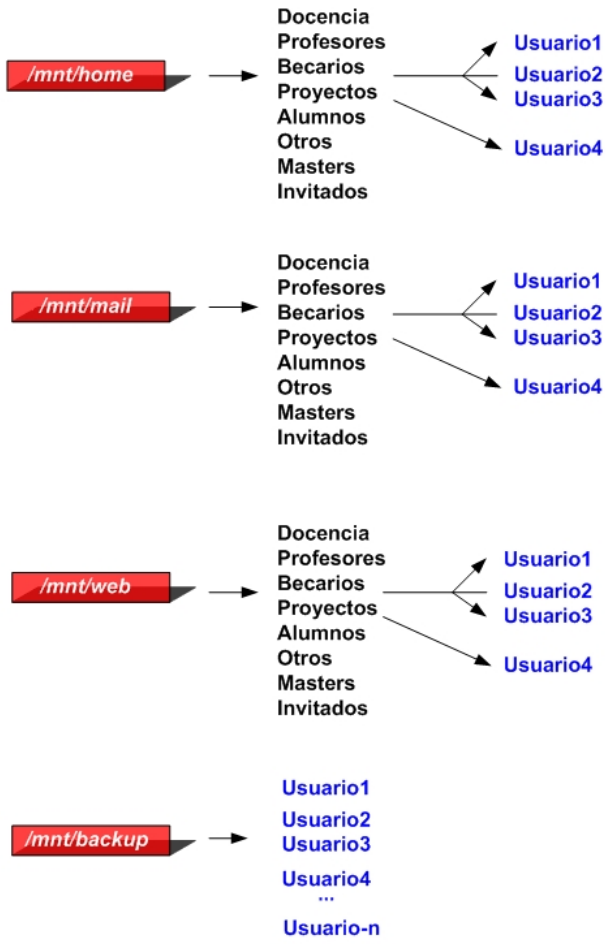


Figura 5.2: Estructura de directorios para las cuentas de usuario

5.3.3. Migración de los directorios de las cuentas de usuario

Una vez creados los grupos, y añadidos los usuarios al directorio Ldap, se necesita crear la estructura de directorios que alojará las cuentas de usuario según los grupos existentes. La estructura aparece reflejada en la figura 5.2.

En *Aguila*, los directorios que contenían las cuentas de usuario estaban en el directorio */export/home/aguila/*, y tenían una estructura igual que el directorio */mnt/home* de la figura 5.2. El nuevo sistema añade los directorios */mnt/mail*, */mnt/web* y */mnt/backup* para separar del *home* de cada cuenta, los directorios para el correo, las páginas web y los datos más críticos.

Desde *Piolin*, en el */mnt/home* del nuevo sistema se sincronizará el directorio */export/home/aguila* de *Aguila*:

```
rsync -au4 --delete aguilaY:/export/home/aguila /mnt/home
```

Se crean los directorios que faltan, y se establecen los permisos:

```
# mkdir /mnt/mail/docencia
# mkdir /mnt/mail/profesores
# mkdir /mnt/mail/becarios
# mkdir /mnt/mail/proyectos
# mkdir /mnt/mail/alumnos
# mkdir /mnt/mail/otros
# mkdir /mnt/mail/masters
# mkdir /mnt/mail/invitados

# mkdir /mnt/mail/docencia
# mkdir /mnt/mail/profesores
# mkdir /mnt/mail/becarios
# mkdir /mnt/mail/proyectos
# mkdir /mnt/mail/alumnos
# mkdir /mnt/mail/otros
# mkdir /mnt/mail/masters
# mkdir /mnt/mail/invitados

# chown root:docencia /mnt/*/docencia
# chown root:profesores /mnt/*/profesores
# chown root:becarios /mnt/*/becarios
# chown root:proyectos /mnt/*/proyectos
# chown root:alumnos/mnt/*/alumnos
# chown root:otros /mnt/*/otros
```

```
# chown root:masters /mnt/*/masters
# chown root:invitados /mnt/*/invitados
```

Conversión de los directorios home de cada cuenta de usuario al nuevo sistema

Los home de usuario de *Aguila* están sincronizados en el directorio */mnt/home* del nuevo sistema. Los siguientes pasos a realizar son:

- Convertir todos los correos de cada cuenta de usuario al formato *Maildir* y depositarlos en */mnt/mail*.
- Mover los directorios *public_html* y *private_html* de cada cuenta de usuario a sus respectivos directorios dentro de */mnt/web*.
- Actualizar varios de los atributos, de cada cuenta de usuario, en el directorio Ldap.

Para automatizar los pasos anteriores, se crearán dos *scripts*: el primero listará el nombre de todas las cuentas de usuario del directorio Ldap y el segundo realizará dichos pasos con cada uno de los nombres listados. Además se necesita instalar el paquete *mb2md*, que convierte correos del formato *mbox* al formato *Maildir*:

```
# apt-get install mb2md
```

El primer *script* (ver página 263.) es una consulta Ldap con un filtro. Se utilizará para generar el fichero *nombres.txt*:

```
# ./listado.sh > nombres.txt
```

El segundo *script* realiza en primer lugar una copia del directorio */var/mail* de *Aguila* en el */var/mail* de *Piolin*. A continuación realiza las siguientes tareas por cada uno de los usuarios que aparecen en el listado generado por el primer script(*nombres.txt*):

- Modifica las rutas *sambaHomePath* ,*sambaHomeDrive*, *sambaLogonScript*, *sambaProfilePath* por *'\\disco\homes'*, *'Z:'*, *'netlogon.bat'* y *'\\disco\profiles'* respectivamente. (Por defecto no se utiliza el recurso *profiles*, pero se introduce en caso de necesidad).
- Cambia la *shell* de usuario por */bin/bash* en el caso de que utilizase otra.
- Modifica la ruta del home antigua (*/export/home/aguila/[grupo]/[nombre de usuario]*) por la nueva (*/mnt/home/[grupo]/[nombre de usuario]*).
- Crea los directorios */mnt/web/[grupo]/[nombre de usuario]* y */mnt/mail/[grupo]/[nombre de usuario]* correspondientes a la cuenta de usuario.

- Convierte los correos de cada cuenta de usuario al formato *Maildir*. Los correos están dispersos en tres localizaciones distintas que el *script* tratará de una en una:

  1. La primera localización es el directorio */var/mail* de *Aguila*, copiado en el */var/mail* de *Piolin*.
  2. La segunda localización es el fichero *mbox*, en el directorio raíz de la cuenta.
  3. La tercera y última localización es el directorio *mail*, en el directorio raíz de la cuenta (localización utilizada por Squirrelmail).

  El script utiliza el comando *mb2md* instalado anteriormente para realizar las conversiones.

- Mueve los directorios *.web* y *.sweb* del directorio *home* de la cuenta a los directorios */mnt/web/[grupo]/[nombre de usuario]/public_html* y */mnt/web/[grupo]/[nombre de usuario]/private_html* respectivamente.

- Crea enlaces simbólicos dentro del directorio */mnt/home* y del directorio */mnt/mail* correspondientes, que apunten a la localización final de los directorios *public_html* y *private_html* (en */mnt/web*).

- Crea un fichero *.htacces* en el directorio *private_html* que restringe el acceso a la web privada de la cuenta. Solicita autenticación y sólo da permiso al propio usuario para acceder a la web segura. Este fichero se crea por seguridad, pero puede ser modificado posteriormente por el usuario para adecuarlo a sus necesidades.

- Por último, crea el directorio */mnt/backup/[nombre de la cuenta]* y un enlace simbólico que apunta a dicho directorio en el home de la cuenta.

Una vez ejecutados los dos *scripts*, las cuentas del antiguo *Aguila* están perfectamente migradas al nuevo sistema, salvo las contraseñas. En el apartado siguiente se explica como se solucionó la incidencia.

5.3.4. Sincronización de contraseñas en el directorio Ldap

Las contraseñas de los usuarios se guardan en el directorio Ldap de dos formas distintas para su utilización en Linux y en Windows, pero al migrar las cuentas de *Aguila* al nuevo sistema, solo se migra la contraseña para Linux, debiendo volver a introducirse si se quiere utilizar en Windows. Para paliar este inconveniente, se avisó a los usuarios de que se conectasen a *Caponata* (servidor SSH) con su contraseña de Linux y volviesen a introducirla.

En *Caponata* se ha sustituido el comando *passwd* por el comando *smbpasswd*, creando un enlace simbólico para que los usuarios puedan seguir utilizando el comando *passwd*. Cuando un usuario hace uso del comando *passwd*, como en el siguiente ejemplo (usuario *folcina*):

```
$ passwd
Old SMB password: xxxxxx
New SMB password: xxxxxx
Retype new SMB password: xxxxxx
Password changed for user folcina
```

el servidor Samba instalado en *Caponata*, se conecta con el servidor Ldap en *Piolin* y modifica los atributos que contienen las contraseñas de Linux y de Windows, de manera que todos ellos contengan la misma contraseña, pero cada uno con el cifrado correspondiente. De esa forma, se mantienen siempre sincronizadas las contraseñas.

Finalmente, se puede comprobar como un usuario que ha actualizado su contraseña desde *Caponata*, tiene actualizada toda la información en el servidor Ldap:

```
# smbldap-usershow folcina
dn: uid=folcina,ou=People,dc=arcos,dc=inf.uc3m.es
objectClass: posixAccount,inetOrgPerson,shadowAccount,sambaSamAccount
uid: folcina
cn: Francisco Olcina Grande
sn: Grande
uidNumber: 3021
gidNumber: 3000
gecos: Francisco Olcina Grande,,,
shadowMax: 99999
shadowWarning: 7
sambaSID: S-1-5-21-3492619381-3135118558-3133272105-7042
loginShell: /bin/bash
homeDirectory: /mnt/home/becarios/folcina
sambaProfilePath: \\lucas\profiles
sambaLogonScript: netlogon.bat
sambaHomeDrive: Z:
sambaHomePath: \\lucas\homes
sambaPasswordHistory: 00000000000000000000000000000000
shadowLastChange: 13459
sambaKickoffTime: 1230764399
sambaLMPassword: AC85xxxxxxxxxxxxxxxx
sambaNTPassword: 303xxxxxxxxxxxxxxxxx
```

```
sambaPwdMustChange: 2147483647
sambaAcctFlags: [U          ]
mail: folcina@arcos.inf.uc3m.es
sambaPwdCanChange: 1189549061
sambaPwdLastSet: 1189549061
userPassword: {SMD5}hWxxxxxxxxxxxxxxxxxx
```

5.3.5. Creación de scripts

Un último paso para una correcta migración de las cuentas de usuario del antiguo sistema al nuevo, es la creación de un sistema automático de creación de usuarios. En el servidor *Aguila* del antiguo sistema, se utilizaba un *script* para la creación de usuarios. Éste *script* utilizaba el comando *adduser* y creaba los directorios *.web* y *.sweb* (*public_html* y *private_html* respectivamente). En el nuevo sistema, la creación de un usuario nuevo implica los siguientes pasos:

- Crear una entrada en el directorio Ldap con la información de dicho usuario:
  - Nombre completo.
  - Nombre de usuario.
  - *Uid* del usuario (se busca uno automáticamente).
  - Grupo del usuario (según los grupos listados en la página 129.).
  - Utilización de la *shell /bin/bash*.
  - Ruta del directorio *home* (*/mnt/home/[grupo de la cuenta de usuario]/[nombre de la cuenta de usuario]*).
  - Ruta del directorio *home* para Samba (*'\\disco\homes'*).
  - Ruta de la unidad en Windows que será el directorio *home*. (*'Z:'*).
  - Nombre del *script* de inicio en Windows (*netlogon.bat*).
  - Ruta del profile de Windows (*'\\disco\profiles'*, por defecto no se utilizará este recurso).
- Introducir una contraseña para el mismo utilizando *smbldap-passwd*. Éste comando, de las Smbldap-Tools, actualiza la contraseña en el directorio Ldap, tanto para Windows, como para Linux.
- Crear el directorio */mnt/mail/[grupo de la cuenta de usuario]/[nombre de la cuenta de usuario]*.
- Crear los directorios */mnt/web/[grupo de la cuenta de usuario]/[nombre de la cuenta de usuario]/public_html* y */mnt/web/[grupo de la cuenta de usuario]/[nombre de la cuenta de usuario]/private_html*.

- Crear el directorio */mnt/backup/[nombre de la cuenta de usuario].*
- Crear los enlaces simbólicos necesarios en */mnt/home/[grupo de la cuenta de usuario]/[nombre de la cuenta de usuario]* y */mnt/mail/[grupo de la cuenta de usuario]/[nombre de la cuenta de usuario].*
- Establecer los permisos adecuados en los directorios */mnt/home/...* , */mnt/mail/...* y */mnt/web/...*
- Crear el fichero *.htaccess* en el directorio *private_html* de la cuenta de usuario, para restringir el acceso via web mediante autenticación, permitiendo el acceso únicamente al propietario de la cuenta.
- Enviar un correo a la cuenta de usuario para que se creen automáticamente los directorios necesarios para el formato *Maildir* dentro del directorio */mnt/mail/[grupo de la cuenta de usuario]/[nombre de la cuenta de usuario]/Maildir.*

Todos los pasos anteriores los realiza el *script cuentas.sh* dentro de *Piolin* (ver página 265.), que incorpora un interfaz gráfico en modo texto.

Por otro lado, el *script listado.sh* (ver página 263.), creado en el proceso de migración, será útil en cualquier tarea que requiera un listado de los nombres de las cuentas que aparecen en el directorio Ldap, en un momento determinado.

5.4. Fase IV: instalación y configuración del resto de servicios

5.4.1. Servicio web

El servidor web principal del sistema funcionará en *Piojito*, mediante Apache y Apache-ssl. En el DNS aparecerá piojito como máquina principal del dominio *arcos.inf.uc3m.es*, consiguiendo que *Piojito* sirva la página principal del dominio. La página principal de *Piojito*, será la página del grupo ARCOS en inglés, alojada en una cuenta de usuario. Las funciones que debe cumplir el servidor aparecen en el capítulo de diseño (ver página 108.).

Para realizar la instalación, se ejecutan los siguientes comandos en *Piojito*:

```
# apt-get install apache apache-ssl libapache-auth-ldap php4  php4-common
```

Una vez instalados los paquetes anteriores, se configura el fichero */etc/apache/httpd.conf* (ver página 267.) y el fichero */etc/apache-ssl/httpd.conf* (ver página 276.).

Para la página web principal se utiliza la cuenta de usuario *'web'*, que se crea mediante el *script cuentas.sh* (ver página 136.). La cuenta *'web'* contará con los directorios *public_html* y *private_html* que serán las direcciones web *http://arcos.inf.uc3m.es* y *https://arcos.inf.uc3m.es* respectivamente.

Por último, se inician los servidores Apache y Apache-ssl:

```
# /etc/init.d/apache start
# /etc/init.d/apache-ssl start
```

5.4.2. Servicio de correo

El servidor de Correo principal del sistema funcionará en *Piojito*, mediante Postfix. Piojito realizará las tareas de servidor Smtp, Pop3 e Imap. El resto de las máquinas también utilizarán Postfix y tendrán a *Piojito* como máquina predeterminada para hacer *relay*. Para la instalación de Postfix en todas las máquinas que componen el sistema (*Donald, Daisy, Boyerito, Piojito, Piolin, Lucas* y *Caponata*), se instala el paquete Postfix:

```
# apt-get install postfix
```

En el caso de *Piojito*, se instalan también los siguientes paquetes:

```
# apt-get install spamassassin clamav amavisd-new courier-imap \
                courier-imap-ssl courier-pop  courier-pop-ssl
```

El fichero de configuración de Postfix es */etc/postfix/main.cf*. *Piojito* tiene su fichero *main.cf* (ver página 285.) configurado para utilizar los *Maildir*, asi como Spamassasin, Clamav y Amavis. Además, en *Piojito* hay que añadir las siguientes líneas al fichero */etc/postfix/master.cf*:

```
smtp-amavis     unix    -       -       n       -       2       smtp

        -o smtp_data_done_timeout=1200



127.0.0.1:10025 inet    n       -       n       -       -       smtpd

        -o content_filter=

        -o local_recipient_maps=

        -o relay_recipient_maps=

        -o smtpd_restriction_classes=
```

```
-o smtpd_client_restrictions=

-o smtpd_helo_restrictions=

-o smtpd_sender_restrictions=

-o smtpd_recipient_restrictions=permit_mynetworks,reject

-o mynetworks=127.0.0.0/8

-o strict_rfc821_envelopes=yes

-o smtpd_error_sleep_time=0

-o smtpd_soft_error_limit=1001

-o smtpd_hard_error_limit=1000
```

El resto de máquinas, poseen un fichero */etc/postfix/main.cf* equivalente (ver página 286.), salvo en la línea *myhostname*, donde irá indicado el nombre de la máquina concreta.

Por último, en cada una de las máquinas se creará el fichero */etc/mailname* con el nombre completo de la máquina (incluido el dominio) y se iniciará el servidor Postfix:

```
# echo ‘‘[nombre de la máquina].arcos.inf.uc3m.es > /etc/mailname
# /etc/init.d/postfix start

( en el caso de Piojito )
# /etc/init.d/courier-imap start
# /etc/init.d/courier-imap-ssl start
# /etc/init.d/courier-pop start
# /etc/init.d/courier-pop-ssl start
```

5.4.3. Servicio de resolución de nombres (DNS)

La resolución de nombres se realizará en el sistema mediante el fichero */etc/hosts* y DNS. El DNS principal del dominio ARCOS estará alojado en *Piojito*, utilizando Bind9. Para su instalación, se ejecuta en *Piojito*:

```
# apt-get install bind9
```

Después se configura el fichero */etc/bind/named.conf* (ver página 287.) y los ficheros */etc/bind/db.arcos.inf.uc3m.es* (ver página 288.) y */etc/bind/db.148.117.163* (ver página 289.).

Para todas las máquinas, se debe configurar el fichero */etc/resolv.conf* (ver página 290.) para que utilice el DNS de *Piojito* y como secundario el de la Universidad (163.117.131.31s). Por último, se configura en todas las máquinas el fichero */etc/hosts* (ver página 290.) para que contenga las direcciones IP y nombres de las máquinas que componen el sistema. De esta forma, no tienen que utilizar el servidor DNS para conocer la *ip* de las otras máquinas, siendo la comunicación más rápida.

5.4.4. Servicio de gestión de bases de datos (MySQL)

El servidor MySQL del sistema, que gestionará las bases de datos que se necesiten, se alojará en *Piojito*. Para realizar una gestión más cómoda de las bases de datos, se instalará el gestor via web Phpmyadmin.

En piojito, se ejecuta:

```
# apt-get install  mysql-server-4.1  mysql-client-4.1 php5.0-mysql phpmyadmin
# /etc/init.d/mysql start
# /etc/init.d/apache-ssl restart ( para poder utilizar el phpmyadmin )
```

Por último, se establece la contraseña del administrador de MySql:

```
# mysqladmin -u root password [contraseña]
```

5.4.5. Servicio de terminal remoto (SSH)

La máquina *Caponata* servirá como terminal remoto Linux mediante el servidor SSH que trae incorporado. Ésta máquina, además del servicio SSH, contempla una serie de restricciones de seguridad (ver página 113.), que utilizan la librería *snoopy* y el fichero */etc/security/limits.conf* (ver página 290.).

Para instalar la librería *snoopy* se ejecutan en *Caponata* los siguientes comandos:

```
# apt-get install snoopy
# echo '/lib/snoopy.so' >> /etc/ld.so.preload
```

5.4.6. Configuración de Windows para utilizar el dominio Samba ARCOS

El nuevo sistema ofrece la posibilidad de añadir una máquina con Microsoft Windows instalado al dominio Samba denominado ARCOS. A continuación se mostrarán los pasos para introducir una máquina con Microsoft Windows Xp recién instalado al dominio ARCOS.

En primer lugar se desactivan los perfiles móviles, dado que una vez unida la máquina al dominio, cuando un usuario del dominio inicie sesión, se desea que tenga un perfil local. Para permitir solo los perfiles locales hay que seguir los siguientes pasos:

- En *inicio/ejecutar* se escribe: *gpedit.msc*
- Aparecerá el cuadro de directivas de grupo de Windows. En él hay que ir al apartado: *configuración del equipo/plantillas administrativas/sistema/perfiles de usuario.*
- A la derecha de la pantalla aparecen la directivas del grupo seleccionado a la izquierda. Se selecciona la directiva *permitir sólo perfiles de usuario locales*, y se habilita.

La figura 5.3 muestra la directiva ya habilitada. A continuación, se pulsa con el boton

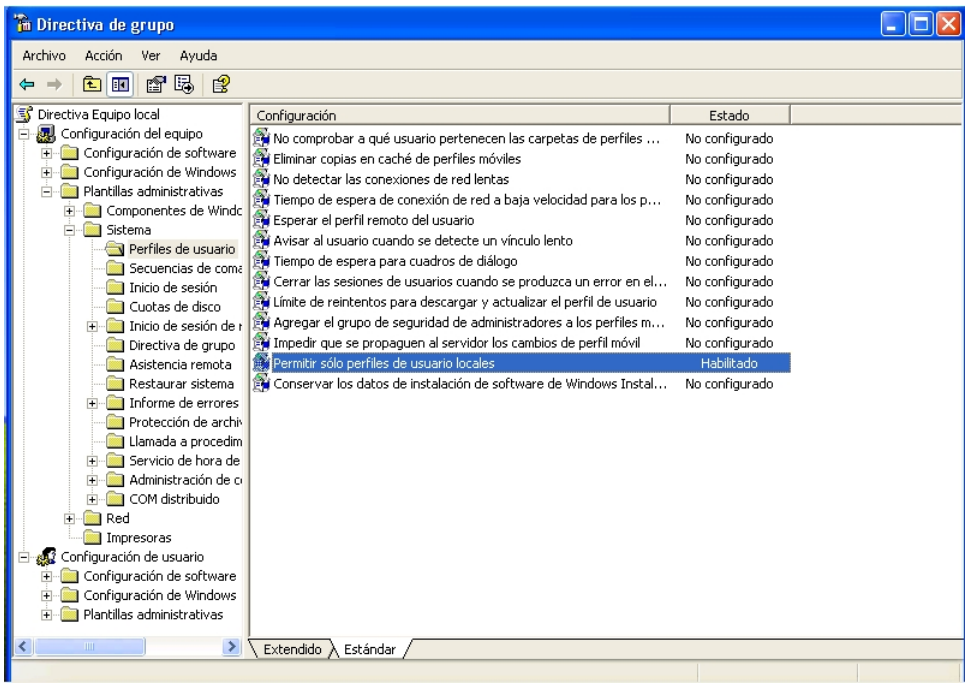


Figura 5.3: Directiva para habilitar únicamente los perfiles locales en Windows

derecho del ratón sobre el icono *Mi Pc* y se accede al panel de propiedades del mismo. En el panel, se accede a la pestaña *Nombre del equipo*, como aparece en la figura 5.4.

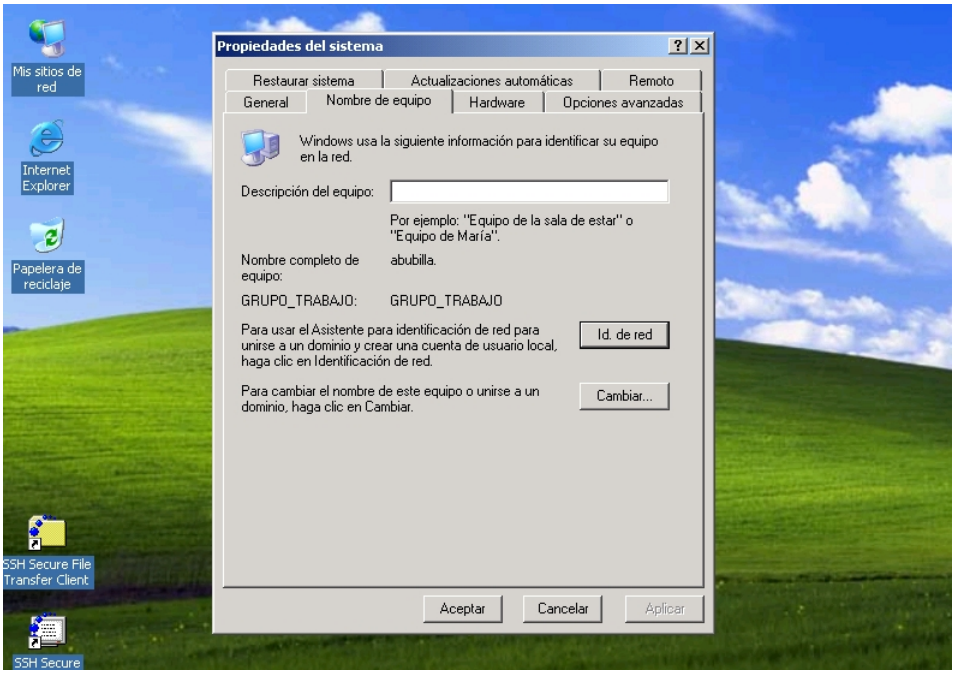


Figura 5.4: Panel de propiedades de *Mi Pc*

Dentro de la pestaña *Nombre del equipo*, se pulsa sobre *Id. de red*, accediendo al *asistente para identificación de red*, como se ve en la figura 5.5. Al pulsar en el asistente sobre *siguiente*, se accede a otra pantalla donde hay que seleccionar la opción *el equipo forma parte de una red organizativa y lo utilizo para conectarme a otros equipos en el trabajo*, como muestra la figura 5.6.

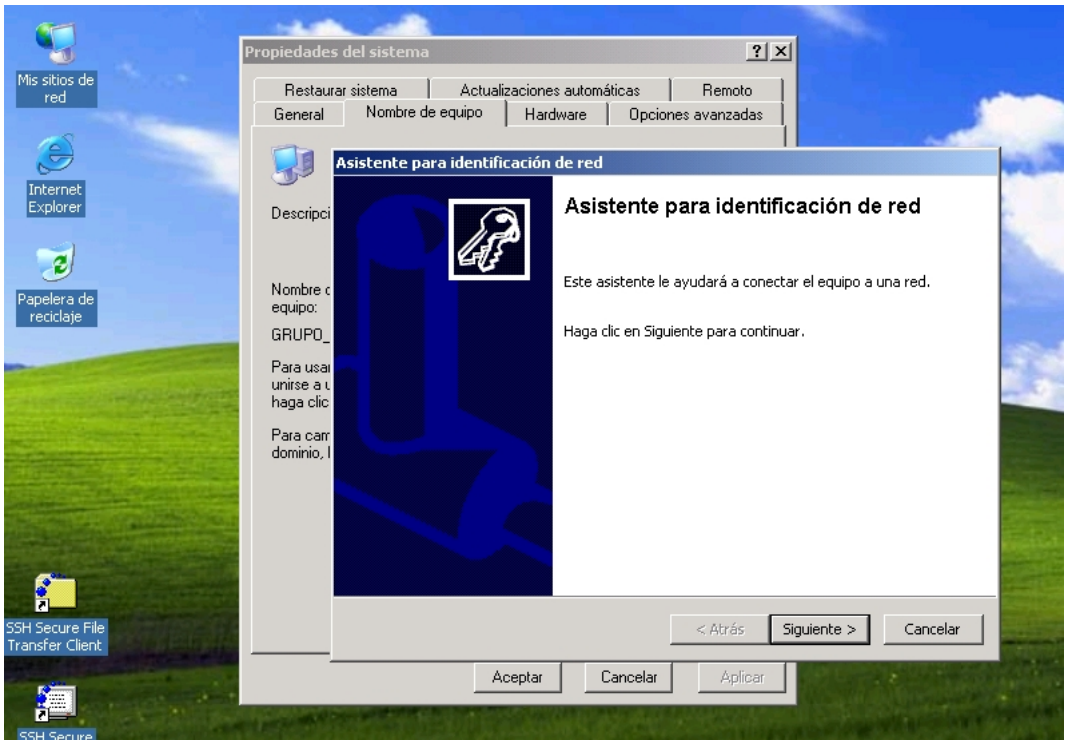


Figura 5.5: *Asistente para identificación de red* de Windows

Se pulsa otra vez sobre *siguiente* y en la pantalla que aparece se selecciona la opción mi compañía utiliza una red con dominio, como muestra la figura 5.7. Al pulsar otra vez sobre siguiente aparece una pantalla donde aparecen los siguientes datos: nombre de usuario, contraseña y dominio, según muestra la figura 5.8. En esta pantalla no se debe hacer nada, por tanto se pulsa sobre siguiente. En la siguiente pantalla, hay que escribir el nombre del dominio (ARCOS) en el apartado *dominio del equipo*, como muestra la figura 5.9.

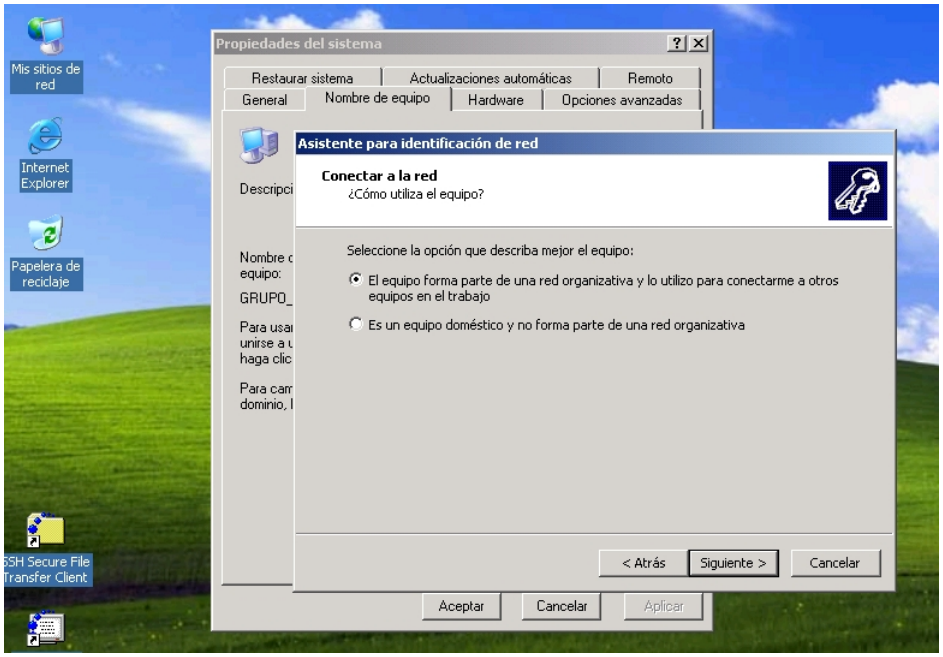


Figura 5.6: Apartado de *¿Cómo utiliza el equipo?* en el *asistente para identificación de red* de Windows

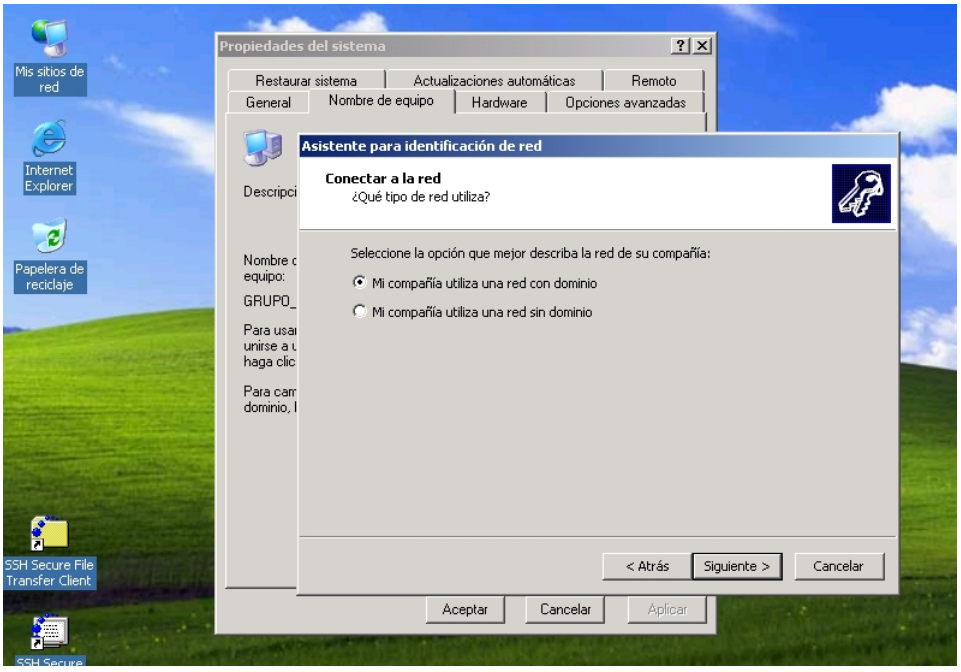


Figura 5.7: Apartado de *¿Qué tipo de red utiliza?* en el *asistente para identificación de red* de Windows

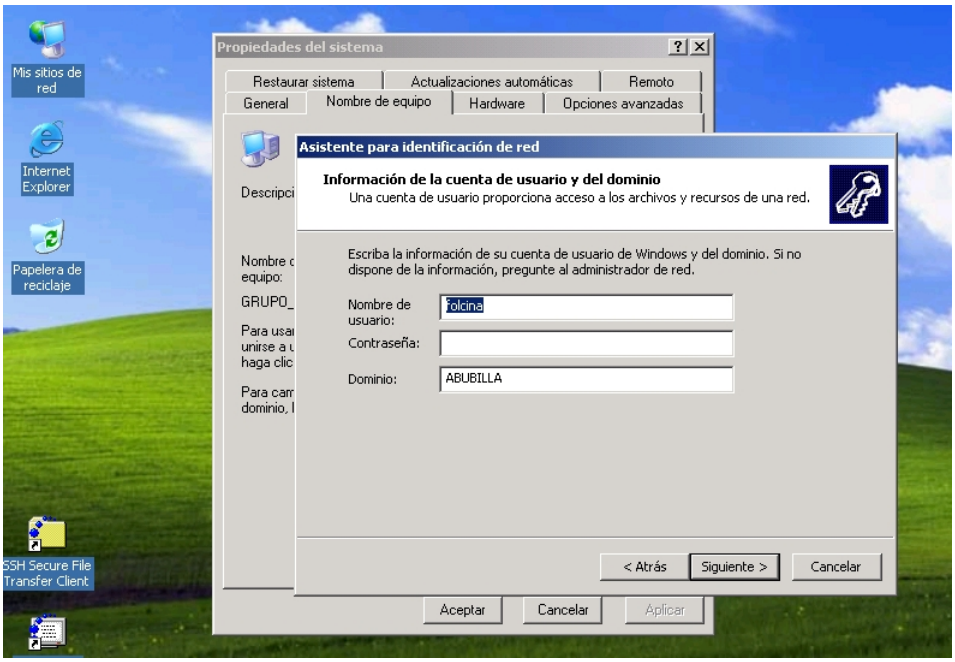


Figura 5.8: Apartado de información de la cuenta de usuario y del dominio en el *asistente para identificación de red* de Windows

Al pulsar sobre siguiente aparece un cuadro donde se escribe el nombre de usuario, la contraseña y el dominio de un usuario con privilegios para unir la máquina al dominio. Se utiliza la cuenta de *administrator* creada por las Smbldap-Tools para unir la máquina al dominio Samba ARCOS, dado que esta cuenta tiene privilegios de administrador. En la figura 5.10 aparece el cuadro donde introducir los datos de la cuenta *administrator*.

Se pulsa sobre aceptar y se produce la unión de la máquina al dominio. Después de unos segundos aparece una pantalla que permite agregar un usuario al dominio, como muestra la figura 5.11. Se selecciona *no agregar un usuario ahora* y se pulsa sobre siguiente.

La siguiente pantalla indica que ha finalizado el asistente para identificación de red, como muestra la figura 5.12. Por tanto se pulsa sobre finalizar y se reinicia la máquina.

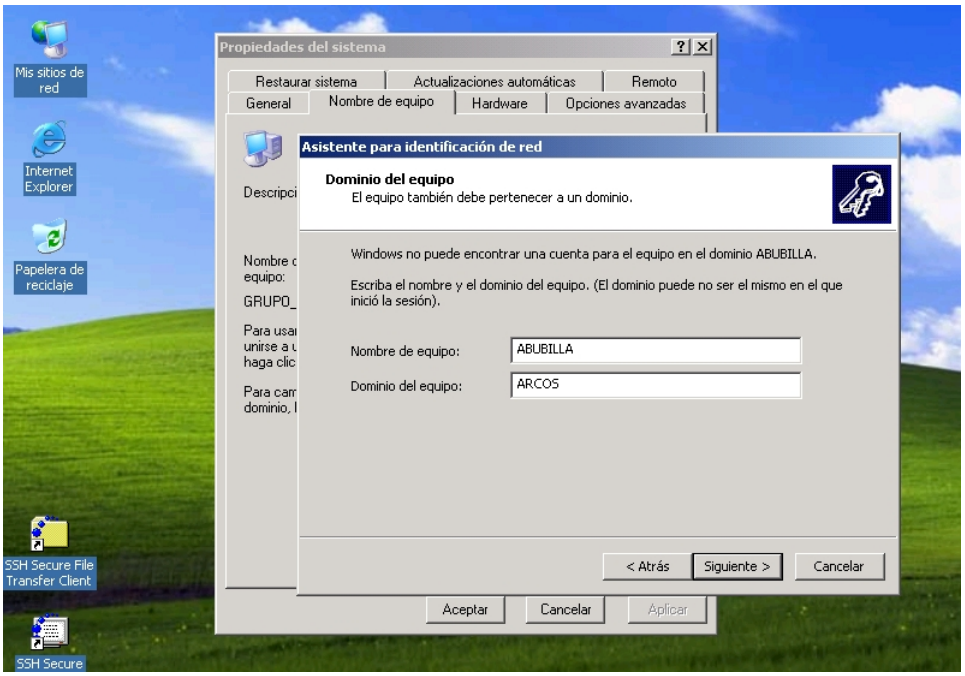


Figura 5.9: Apartado de *dominio del equipo* en el *asistente para identificación de red* de Windows

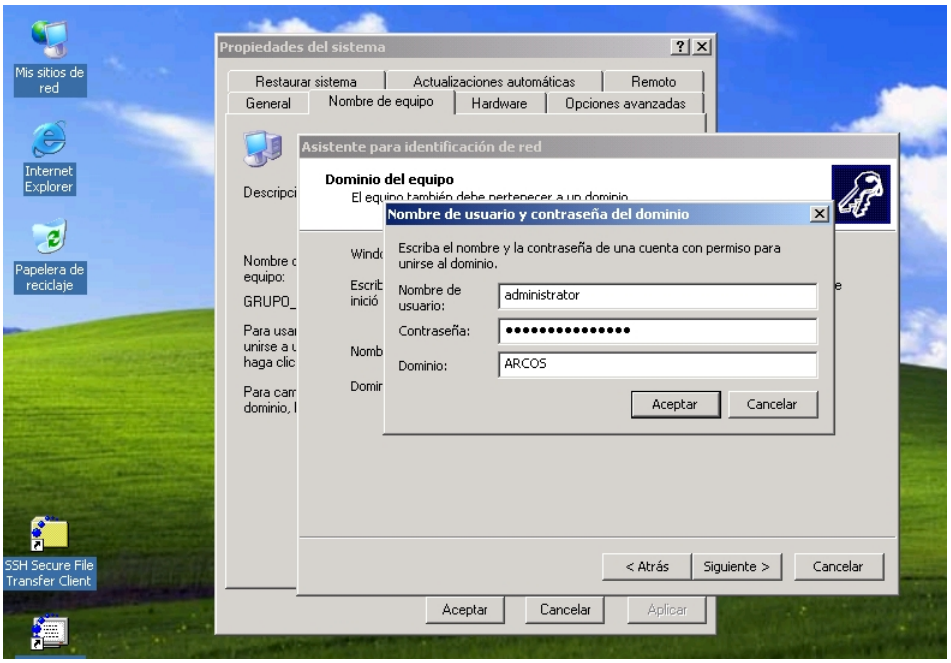


Figura 5.10: Cuadro para introducir los datos de un usuario con privilegios de administración en el dominio

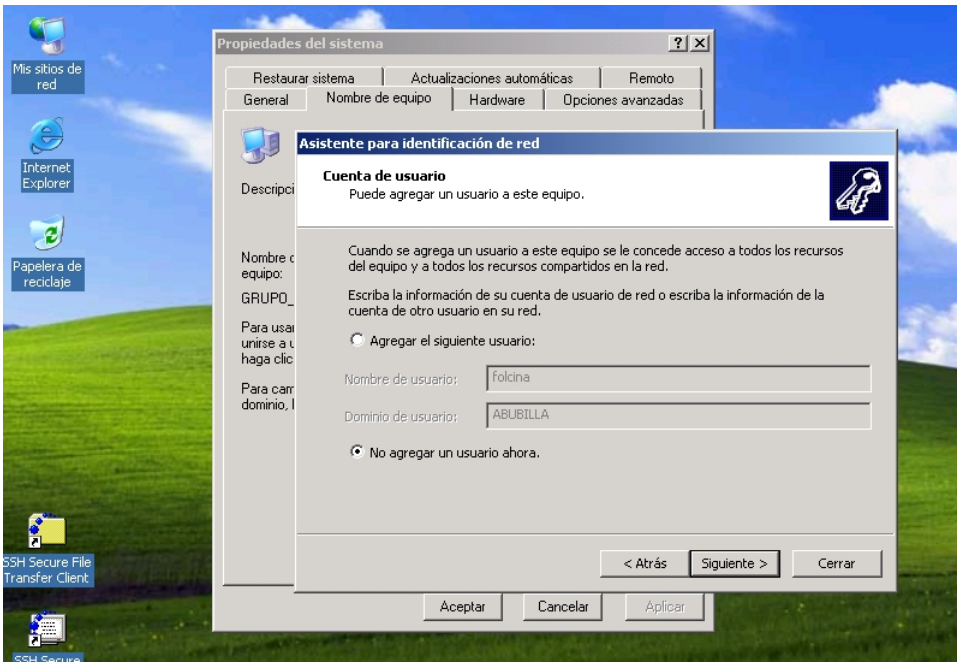


Figura 5.11: Apartado para agregar una cuenta de usuario al dominio

Mientras la máquina se reinicia, se puede comprobar como se ha creado una entrada en el directorio Ldap con la información de la máquina introducida en el dominio. Para ello se ejecuta desde *Piolin* lo siguiente:

```
# smbldap-usershow abubilla$
dn: uid=abubilla$,ou=People,dc=arcos,dc=inf.uc3m.es
objectClass: top,person,organizationalPerson,inetOrgPerson,posixAccount,
             sambaSamAccount
cn: abubilla$
sn: abubilla$
uid: abubilla$
uidNumber: 1018
gidNumber: 515
homeDirectory: /dev/null
loginShell: /bin/false
description: Computer
gecos: Computer
sambaSID: S-1-5-21-3492619381-3135118558-3133272105-3036
sambaPrimaryGroupSID: S-1-5-21-3492619381-3135118558-3133272105-515
displayName: ABUBILLA$
sambaPwdCanChange: 1189695675
```

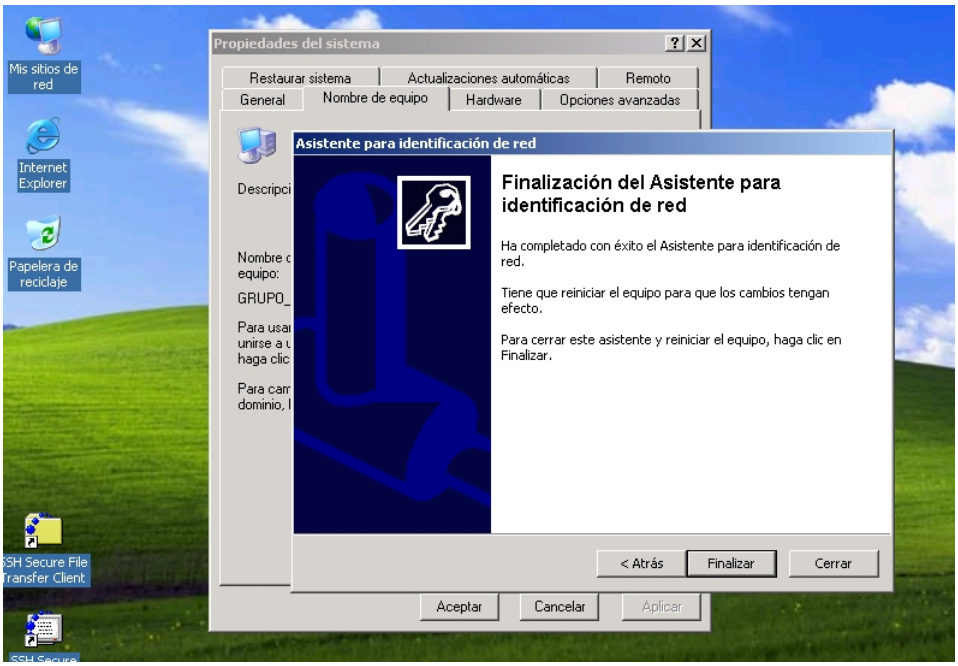


Figura 5.12: Finalización del *asistente para identificación de red* de Windows

```
sambaPwdMustChange: 2147483647
sambaNTPassword: F772ABD771B6E94CEAC1020B4F94AD7A
sambaPwdLastSet: 1189695675
sambaAcctFlags: [W          ]
```

Las máquinas Windows que se unan al dominio Samba ARCOS aparecerán en el directorio Ldap con su nombre seguido de un simbolo '$', como es el caso de *abubilla*.

Una vez que la máquina se ha reiniciado, ya se puede seleccionar en la pantalla de inicio de sesión el dominio ARCOS, como muestra la figura 5.13. Se inicia sesión con una cuenta de usuario del dominio (*folcina*), al ser el primer inicio de sesión, se crea en la máquina un perfil local con la configuración por defecto de Windows. Se puede comprobar en la figura 5.14, como al pulsar sobre el botón de Inicio, aparece el nombre de la cuenta de usuario en la parte superior del menú. Además, en la figura 5.15,se observa como al pulsar sobre Mi Pc, aparecen las unidades de red *H:* y *Z:* correspondientes al directorio *home* de la cuenta y el directorio *BACKUP* respectivamente. Esto se produce por la ejecución del *script netlogon.bat* al iniciar sesión (ver página 124.).

La máquina *abubilla* esta perfectamente unida al dominio Samba ARCOS y en ella puede iniciar sesión cualquier usuario del directorio Ldap de ARCOS.

Figura 5.13: Pantalla de inicio de sesión de Windows Xp

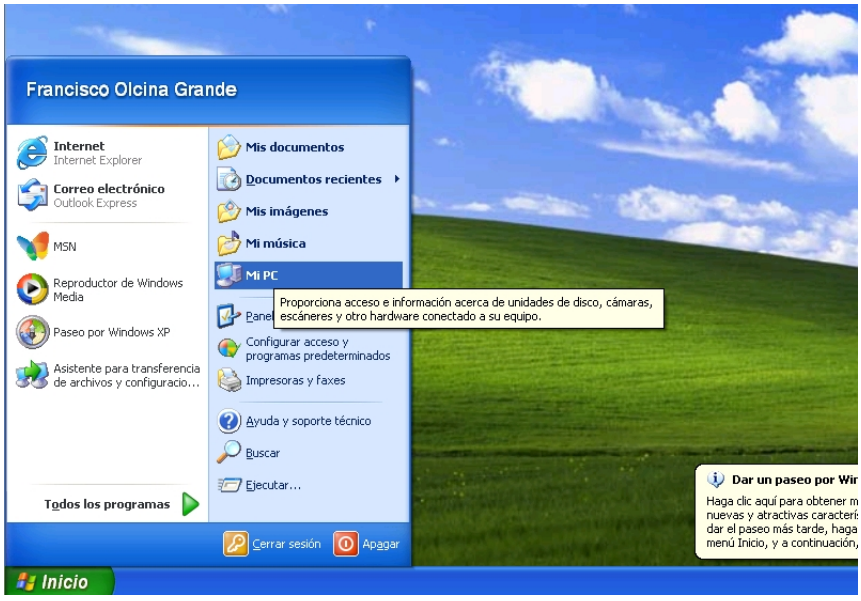


Figura 5.14: Menú de inicio de Windows

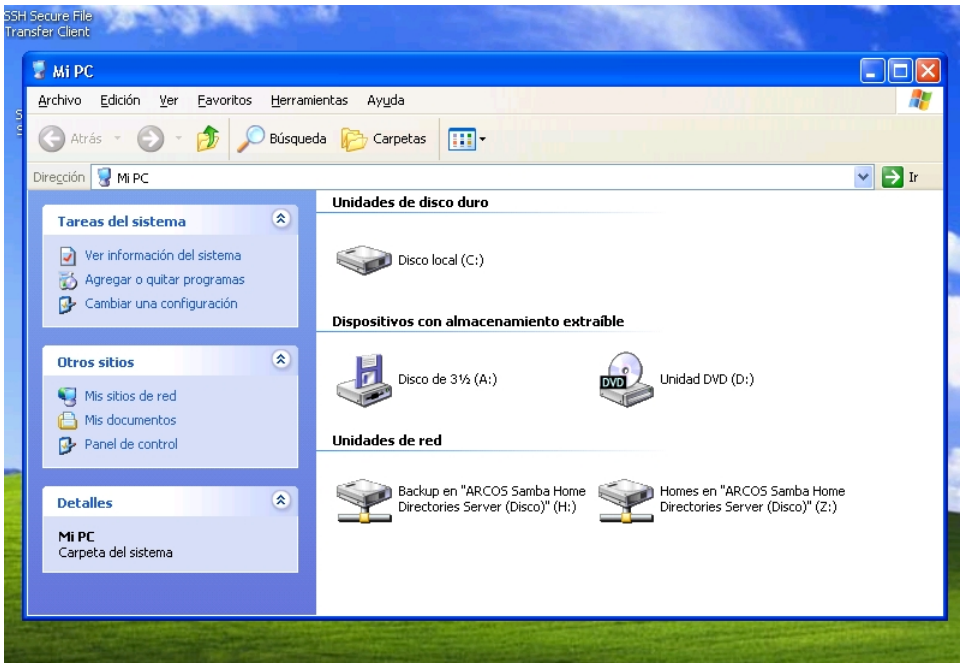


Figura 5.15: Unidades de red del usuario

5.4.7. Creación de una *Intranet*

La *intranet* de ARCOS aparecerá en la dirección *https://arcos.inf.uc3m.es*. Para ello, se utiliza la cuenta *'web'* y su directorio *private_html* correspondiente. En dicho directorio habrá un fichero *.htaccess* generado automáticamente al crear el usuario *'web'* mediante el *script cuentas.sh* (ver página 136.).

El primer paso será adaptar el fichero *.htaccess* (ver página 291.) para que pida autenticación al acceder a la *intranet*, siendo cualquier usuario de ARCOS válido.

El segundo paso consiste en crear las páginas web con enlaces a los distintos servicios que ofrece la *intranet*, dividiendo la *intranet* en dos áreas principales: una para los administradores y otra para el resto de usuarios.

El principal uso de la *intranet* será el gestor de correo web Squirrelmail. Para llevar a cabo su instalación se ejecuta en *Piojito*:

```
# apt-get install squirrelmail
```

Después se ejecuta el *script /etc/squirrelmail/conf.pl*, apareciendo un menú en modo texto, mostrado a continuación:

```
SquirrelMail Configuration : Read: config.php (1.4.3)
```

```
--------------------------------------------------------
Main Menu --
1.  Organization Preferences
2.  Server Settings
3.  Folder Defaults
4.  General Options
5.  Themes
6.  Address Books (LDAP)
7.  Message of the Day (MOTD)
8.  Plugins
9.  Database


D.  Set pre-defined settings for specific IMAP servers

C.  Turn color on
S   Save data
Q   Quit

Command >>
```

Se selecciona la opción 1, apareciendo el siguiente menú, donde se modifican las opciones según se muestra:

```
SquirrelMail Configuration : Read: config.php (1.4.3)
--------------------------------------------------------
Organization Preferences
1.  Organization Name      : ARCOS
2.  Organization Logo      : http://www-es.arcos.inf.uc3m.es/imagenes/arcos.jpg
3.  Org. Logo Width/Height : (165/69)
4.  Organization Title     : SquirrelMail $version (arcos.inf.uc3m.es).
5.  Signout Page           :
6.  Default Language       : es_ES
7.  Top Frame              : _top
8.  Provider link          : http://www.squirrelmail.org/
9.  Provider name          : SquirrelMail

R   Return to Main Menu
C.  Turn color on
S   Save data
Q   Quit
Command >>
```

Se vuelve al menú principal y a continuación se selecciona el submenú de opciones de servidores. Dentro del mismo, se configuran las opciones según aparecen a continuación:

```
SquirrelMail Configuration : Read: config.php (1.4.3)
---------------------------------------------------------
Server Settings

General
-------
1.  Domain                 : arcos.inf.uc3m.es
2.  Invert Time            : false
3.  Sendmail or SMTP       : Sendmail

A.  Update IMAP Settings   : localhost:143 (uw)
B.  Change Sendmail Config : /usr/sbin/sendmail

R   Return to Main Menu
C.  Turn color on
S   Save data
Q   Quit

Command >>
```

Se vuelve al menú principal y se selecciona la opción 8 para habilitar ciertos *plugins* necesarios, siendo el resultado final el que aparece a continuación:

```
SquirrelMail Configuration : Read: config.php (1.4.3)
---------------------------------------------------------
Plugins
  Installed Plugins
    1. fortune
    2. undelete
    3. compose_chars
    4. extract
    5. get_uuencode
    6. download_all
    7. squirrelspell
    8. filters
    9. view_as_html
    10. dictionary-0.6
    11. gpg
    12. html_mail
    13. message_details
    14. info
    15. spamcop
    16. vkeyboard
    17. msg_flags
```

```
    18. preview_pane
    19. archive_mail
    20. folder_sizes
    21. abook_import_export
    22. notes
    23. select_range
    24. notify


  Available Plugins:
    25. administrator
    26. bug_report
    27. listcommands

R   Return to Main Menu
C.  Turn color on
S   Save data
Q   Quit

Command >>
```

Finalmente se pulsa 'S' para grabar los cambios y a continuación 'Q' para salir. El *script* generará el fichero de configuración de Squirrelmail, y se podrá acceder a la página de entrada de Squirrelmail desde *https://arcos.inf.uc3m.es/squirrelmail*, o bien, desde el enlace existente en la *intranet*. Ver figura 5.16 (WebMail).

Figura 5.16: Vista actual de la página principal de la *intranet* de ARCOS

5.5. Fase V: implantación del sistema de *backup* software

Esta fase implica la creación de un sistema de *backup* mediante *scripts* en *bash*. El objetivo de estos *scripts*, se explicó en el capítulo de diseño (ver página 103). A continuación se detallará la instalación de dicho sistema de *backup*.

5.5.1. Creación de *scripts*

En primer lugar, se crea el *script* que comprueba la conectividad con una máquina (*conectividad.sh*, ver página 291.), terminando su ejecución con cero si pudo realizar una conexión por SSH, o uno, si no pudo realizar la conexión por SSH a la máquina especificada como primer parámetro.

A continuación, se crea el *script* de *backup* de máquinas (*backup_raiz.sh*, ver página 292.), que utiliza el *script* de conectividad. Se necesita un fichero *exclude_raiz*, en el que se introducirán los patrones de nombres de ficheros a evitar en la sincronización: *caponata.img*, *lucas.img*, *piojito.img*, *piolin.img* y */maquinas/*/swap**. Éstos nombres corresponden a los ficheros imagen de las máquinas virtuales y a los ficheros de *swap*. No se desea realizar una copia directa de dichos ficheros, ya que el *script backup_raiz.sh* se encarga de realizar una copia de sus sistemas raíz.

El *script backup_raiz.sh* necesita que todas las máquinas de las que vaya a realizar un *backup* tengan en su fichero */etc/fstab* el directorio raiz listo para montar en */mnt/raiz*. El siguiente ejemplo se aplicará a todas las máquinas, cada una con su respectiva partición raíz: (ejemplo en *Lucas*)

```
# echo '/dev/md4  /mnt/raiz   ext3   noauto,defaults  0 0' >> /etc/fstab
```

Después del *script backup_raiz.sh*, se crean los *scripts* de *backup* de los distintos servicios:

- Servicio DNS (*servicio_dns.sh*, ver página 293.).
- Servicio Ldap (*servicio_ldap.sh*, ver página 294.).
- Servicio MySql (*servicio_mysql.sh*, ver página 295.).

Se realizan también *backups* de todos los datos a *Boyerito*, por tanto se crearán tres *scripts* denominados *backup_lucas_a_boyerito_usuarios.sh*, *backup_lucas_a_boyerito_backup_usuarios.sh* y *backup_lucas_a_boyerito_sistema.sh*. Los tres *scripts* son similares, la única variación es la variable *DIR_ORIGEN* y el asunto del *email* que se envía al finalizar el *script*. (Como ejemplo, se adjunta en el Anexo 4, el fichero *backup_lucas_a_boyerito_usuarios.sh*, ver página 297.).

También es necesario realizar el *script* de borrado automático de directorios (*borrado_rotacion.sh*, ver página 298.).

Las salidas de la ejecución de cada *script* se mandan por correo y también se guardan en subdirectorios de */root/salidas* como ficheros. Por tanto, se crean dichos directorios en *Lucas*:

```
# mkdir -p /root/salidas/borrado
# mkdir -p /root/salidas/maquina_daisy
# mkdir -p /root/salidas/maquina_piojito
# mkdir -p /root/salidas/maquina_donald
# mkdir -p /root/salidas/maquina_piolin
# mkdir -p /root/salidas/maquina_caponata
# mkdir -p /root/salidas/maquina_lucas
# mkdir -p /root/salidas/boyerito
# mkdir -p /root/salidas/servicio_ldap
# mkdir -p /root/salidas/servicio_mysql
# mkdir -p /root/salidas/servicio_dns
# mkdir -p /root/temp
```

5.5.2. Ejecución automática

Para la ejecución automática de los *scripts* de *backup*, se hace uso del *crontab* del usuario *root* de *Lucas*, cuyo resultado final es:

```
#############
# MAQUINAS #
#############

# backup raiz piolin
00 2 * * * /root/backup/backup_raiz.sh piolin &> /dev/null

# borrado backup piolin
00 23 * * * /root/backup/borrado_rotacion.sh piolin 60 &> /dev/null

# backup raiz piojito
30 2 * * * /root/backup/backup_raiz.sh piojito &> /dev/null

# borrado backup piojito
05 23 * * * /root/backup/borrado_rotacion.sh piojito 60 &> /dev/null

# backup raiz donald
00 3 * * * /root/backup/backup_raiz.sh donald &> /dev/null
```

```
# borrado backup donald
10 23 * * * /root/backup/borrado_rotacion.sh donald 90 &> /dev/null

# backup raiz daisy
30 3 * * * /root/backup/backup_raiz.sh daisy &> /dev/null

# borrado backup daisy
15 23 * * * /root/backup/borrado_rotacion.sh daisy 90 &> /dev/null

# backup raiz lucas
00 4 * * * /root/backup/backup_raiz.sh lucas &> /dev/null

# borrado backup lucas
20 23 * * * /root/backup/borrado_rotacion.sh lucas 20 &> /dev/null

# backup raiz boyerito
00 4 * * * /root/backup/backup_raiz.sh boyerito &> /dev/null

# borrado backup boyerito
25 23 * * * /root/backup/borrado_rotacion.sh boyerito 60 &> /dev/null

# backup raiz caponata
30 4 * * * /root/backup/backup_raiz.sh caponata &> /dev/null

# borrado backup caponata
30 23 * * * /root/backup/borrado_rotacion.sh caponata 15 &> /dev/null

##############
# SERVICIOS #
##############

# backup servicio mysql
00 5 * * * /root/backup/servicio_mysql.sh &> /dev/null

# borrado backup mysql
35 23 * * * /root/backup/borrado_rotacion.sh mysql 365 &> /dev/null

# backup servicio ldap
15 5 * * * /root/backup/servicio_ldap.sh &> /dev/null

# borrado backup ldap
40 23 * * * /root/backup/borrado_rotacion.sh ldap 365 &> /dev/null

# backup servicio dns
30 5 * * * /root/backup/servicio_dns.sh &> /dev/null

# borrado backup dns
45 23 * * * /root/backup/borrado_rotacion.sh dns 365 &> /dev/null
```

```
#### BACKUP DE LUCAS EN BOYERITO #####
00 6 * * * /root/backup/backup_lucas_a_boyerito_usuarios.sh &> /dev/null
00 7 * * * /root/backup/backup_lucas_a_boyerito_backup_usuarios.sh &> /dev/null
00 8 * * * /root/backup/backup_lucas_a_boyerito_sistema.sh &> /dev/null ; ssh boyerito halt
```

5.5.3. Sincronización entre *Donald* y *Daisy*

El último apartado respecto a las copias de seguridad, es la sincronización que debe existir entre *Donald* y *Daisy*. Para ello, *Donald* utiliza el mismo sistema de sincronización que *Lucas* realiza en *Boyerito*: la utilización de tres *scripts* para cada uno de los volúmenes lógicos existentes (*usuarios*, *backup_usuarios* y *sistema*).

En *Donald*, se crea cada uno de los *scripts*, denominados: *backup_donald_a_daisy_usuarios.sh*, *backup_donald_a_daisy_backup_usuarios.sh* y *backup_donald_a_daisy_sistema.sh*. La única diferencia entre cada uno de ellos es el directorio origen y destino a la hora de realizar la sincronización, y el asunto del *email* enviado. (Se incluye en el Anexo 4 de presente documento el *script backup_donald_a_daisy_usuarios.sh* como ejemplo, ver página 299.).

Se crean los directorios siguientes:

```
# mkdir -p /root/salidas/daisy
# mkdir /root/temp
```

Se introducen en el fichero */etc/fstab* de *Donald* las siguientes líneas:

```
# echo '163.117.148.244:/mnt/usuarios       /mnt/usuarios       nfs ro,noauto,noexec,nodev  0 0' >> /etc/fstab
# echo '163.117.148.244:/mnt/backup_usuarios /mnt/backup_usuarios nfs ro,noauto,noexec,nodev  0 0' >> /etc/fstab
# echo '163.117.148.244:/mnt/sistema        /mnt/sistema        nfs ro,noauto,noexec,nodev  0 0' >> /etc/fstab
```

El fichero */etc/fstab* de *Daisy* cuenta con las siguientes líneas, no siendo necesario añadir más:

```
/dev/vg1/usuarios        /mnt/usuarios        nfs noauto,defaults  0 0
/dev/vg1/backup_usuarios /mnt/backup_usuarios nfs noauto,defaults  0 0
/dev/vg1/sistema         /mnt/sistema         nfs noauto,defaults  0 0
```

Por último, en el *crontab* de *root* de *Donald* se añade lo siguiente:

```
00 0 * * * /root/backup/backup_donald_a_daisy_usuarios.sh &> /dev/null
00 1 * * * /root/backup/backup_donald_a_daisy_backup_usuarios.sh &> /dev/null
30 1 * * * /root/backup/backup_donald_a_daisy_sistema.sh &> /dev/null
```

5.6. Fase VI: instalación de software de monitorización

Para una correcta monitorización del sistema se instalarán varias aplicaciones que permitan realizar un seguimiento de los puntos más importantes. La mayoría de las aplicaciones de monitorización que se van a instalar, hacen uso de la herramienta RRDTool,

que permite la creación de gráficas en función de una base de datos de valores, que se crea mediante *scripts*. Todas estas herramientas mostrarán sus resultados vía web, contiendo la *intranet* de ARCOS enlaces a cada una de ellas (ver figura 5.17).

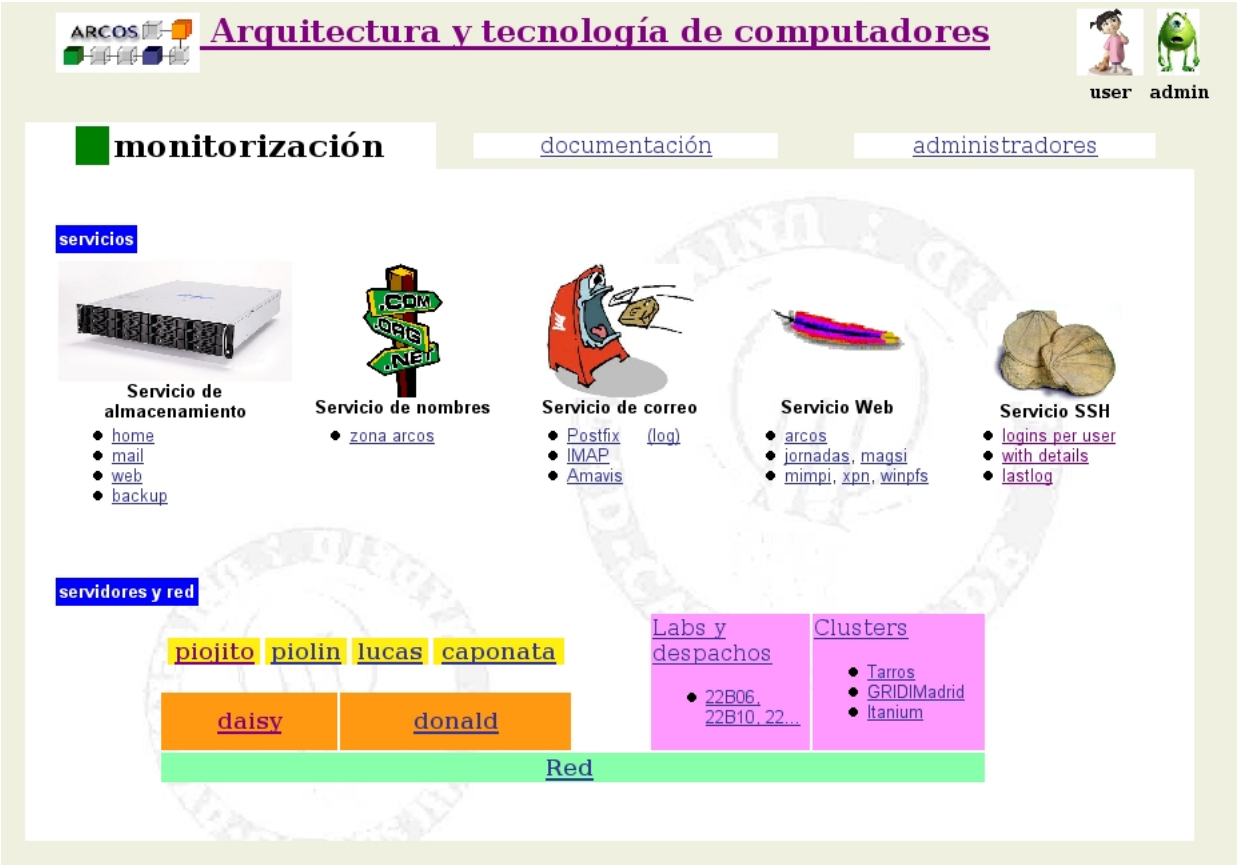


Figura 5.17: Vista actual de la página de monitorización de la *intranet* (en el apartado de administradores)

5.6.1. Instalación de Durep

Durep es un software de monitorización de uso de disco que genera una salida gráfica, más intuitiva que el comando *du* de Linux.

En Lucas, se instala el paquete *durep*:

```
# apt-get install durep
```

Se crea en *Lucas* un *Script* que realiza lo siguiente:

- Genera estadísiticas de disco con dos niveles de profundidad, para los directorios */mnt/home*, */mnt/mail*, */mnt/web* y */mnt/backup* de *Lucas*.
- Envía los ficheros generados (páginas web con gráficos *png*) a la web segura del dominio *arcos.inf.uc3m.es*. Es necesario recordar que la web segura (*https://arcos.inf.uc3m.es*), esta contenida en la cuenta de usuario *'web'*. Por lo tanto, se envían los datos generados al directorio *private_html* de dicha cuenta, utilizando los respectivos subdirectorios: *stats_disk_home, stats_disk_mail, stats_disk_web* y *stats_disk_backup*.
- Borra los ficheros generados en *Lucas*.

Este *script* se guarda como */usr/local/bin/durep.sh* (ver página 300.) en *Lucas* y se introduce en el *crontab*:

```
# du-report
15 14 1 * * /usr/local/bin/durep.sh
```

Las direcciones web donde aparecerán los resultados generados por Durep son las siguientes:

- *https://arcos.inf.uc3m.es/stats_disk_home/*
- *https://arcos.inf.uc3m.es/stats_disk_mail/*
- *https://arcos.inf.uc3m.es/stats_disk_web/*
- *https://arcos.inf.uc3m.es/stats_disk_backup/*

Además existirá un enlace desde la *intranet* de ARCOS a los enlaces anteriores.

5.6.2. Instalación de Bindgraph

Bindgraph es un software de monitorización del servicio DNS que genera gráficas utilizando la herramienta RRDTool. Se ejecuta en el sistema como fichero *cgi*, por lo tanto es necesario un servidor web.

Bindgraph se instala en el sistema en la máquina virtual *Piojito*. Para ello, se instala el paquete *bindgraph*:

```
# apt-get install bindgraph
```

Es necesario que el servidor Bind9 genere ficheros *log* con el resultado de las peticiones realizadas al DNS. Para ello se añade al fichero */etc/bind/named.conf.local* de *Piojito* lo siguiente:

```
logging {
  channel "querylog" { file "/var/log/bind9-query.log"; print-time yes; };
  category queries { querylog; };
};
```

Y a continuación se reinicia el servidor Bind:

```
# /etc/init.d/bind9 restart
```

En el fichero de configuración de Bindgraph se añade lo siguiente:

```
DNS_LOG=/var/log/bind9-query.log
LOG_FORMAT=bind92
```

Y por último se inicia el demonio *bindgraph* en *Piojito*:

```
# /etc/init.d/bindgraph start
```

El enlace que aparece en la *intranet* de ARCOS para acceder al servicio es:
https://arcos.inf.uc3m.es/cgi-bin/bindgraph.cgi

5.6.3. Instalación de Mailgraph

Mailgraph es similar a Bindgraph, se trata de un software de monitorización del uso del servidor de correo de un sistema Linux. Obtiene la información de los ficheros *log* que genera el servidor *smtp* instalado en la máquina, y genera gráficas mediante la herramienta RRDTool. Se ejecuta en el sistema como fichero *cgi*, por lo tanto es necesario un servidor web.

Mailgraph se instala en la máquina virtual *Piojito*:

```
# apt-get install mailgraph
```

Se configura el fichero */etc/default/mailgraph* para que utilice el fichero de *log* utilizado por Postfix, el servidor SMTP instalado en *Piojito*. Para ello, se añaden estas líneas al fichero comentado:

```
MAIL_LOG=/var/log/mail.log
IGNORE_LOCALHOST=true
```

Y por último se inicia el demonio *mailgraph* en *Piojito*:

```
# /etc/init.d/mailgraph start
```

5.6.4. Instalación de Couriergraph

Couriergraph es un software muy similar a Mailgraph, con la diferencia de que genera las estadísticas de uso de los servidores Pop3 e Imap, siempre que se utilice la versión Courier de los mismos. Al igual que Mailgraph, utiliza la herramienta RRDTool para generar estadísticas y se ejecuta en el sistema como fichero *cgi*.

Couriergraph se instala en la máquina virtual *Piojito*:

```
# apt-get install couriergraph
```

Se configura el fichero */etc/default/couriergraph* para que utilice el fichero de *log* utilizado por Courier-pop3 y Courier-imap (incluidas las versiones SSL) en *Piojito*. Para ello, se añade la línea siguiente al fichero comentado:

```
MAIL_LOG=/var/log/mail.log
```

Y por último se inicia el demonio *couriergraph* en *Piojito*:

```
# /etc/init.d/couriergraph start
```

El enlace que aparece en la intranet de ARCOS para acceder al servicio es: *https://arcos.inf.uc3m.es/cgi-bin/mailgraph.cgi*

5.6.5. Instalación de Amavis-stats

Amavis-stats es una herramienta que genera estadísticas sobre el uso de Amavis en un servidor de correo. Utiliza la herramienta RRDTool para generar gráficas, y se ejecuta como una aplicación *php* sobre el sistema.

Amavis-stats se instala en la máquina virtual *Piojito*:

```
# apt-get install amavis-stats
```

Automáticamente se instala en el *crontab* del sistema para que genere las estadísticas de Amavis. Utiliza el fichero */var/log/mail.info* para recopilar la información necesaria.

Amavis-stats introduce el fichero */etc/apache/conf.d/amavis-stats.conf* y */etc/apache-ssl/conf.d/amavis-stats.conf*. Se borra el primer fichero, dado que solo interesa ejecutar Amavis-stats desde la *intranet* de ARCOS, que se sirve desde Apache-ssl. El último paso es reiniciar el servidor Apache-ssl:

```
# /etc/init.d/apache-ssl restart
```

La dirección web donde aparecen los resultados de Amavis-stats es:
https://arcos.inf.uc3m.es/stats_amavis/, existiendo un enlace en la *intranet* de ARCOS.

5.6.6. Instalación de Webalizer

Webalizer es una herramienta que genera estadísticas de uso de un servidor web. Éstas estadísticas las muestra mediante páginas web generadas automáticamente, que a su vez incorporan gráficas en forma de diagramas de barras.

Webalizer se instala en la máquina virtual Piojito:

```
# apt-get install webalizer logresolve
```

Se crea un *script* para realizar las estadísticas de las webs deseadas y generar las webs resultantes en una localización prefijada. La localización será el directorio *stats_apache* dentro del directorio *private_html* de la cuenta *'web'*, donde se aloja la *intranet* del dominio *arcos.inf.uc3m.es*. Se realizarán estadísticas de los distintos dominios existentes en el servidor Apache de ARCOS:

- *arcos.inf.uc3m.es*
- *jornadas.inf.uc3m.es*
- *magsi.inf.uc3m.es*
- *mimpi.inf.uc3m.es*
- *xpn.inf.uc3m.es*
- *winpfs.inf.uc3m.es*

La ruta del *script* es */usr/local/sbin/webalizer.sh* (ver contenido en página 301.), realizando el *script* las siguientes tareas:

- Copia en un directorio temporal las estadísticas generadas anteriormente.
- Genera una copia de los ficheros *log* de Apache de cada dominio, resolviendo las direcciones IP que en ellos aparecen, obteneniendo los nombres de dominio. Para ello utiliza el comando *logresolve2*. De esta forma, webalizer utilizará ficheros *log* de Apache con nombres de *hosts*, generando las estadísticas y mostrando los dominios implicados.
- Por último utiliza el comando *webalizer* para generar las estadísticas, usando un ficheros de configuración por cada dominio virtual. Se incluye como ejemplo, en el anexo 4 del presente documento, el fichero */etc/webalizer-arcos.conf* (ver página 302.), que muestra la configuración de Webalizer para el dominio *arcos.inf.uc3m.es*.
- Copia el directorio temporal donde se han generado las estadísticas al directorio *private_html/stats_apache* de la cuenta *'web'*.

- Borra los ficheros temporales.

En *Piojito*, se introduce en el *crontab* de *root* la siguiente línea:

```
5 1 * *   *       /usr/local/sbin/webalizer.sh > /var/log/apache/webalizer_log 2>&1
```

Como resultado de la instalación, existe una web con los datos de Webalizer por cada uno de los dominios virtuales, estando dichos enlaces en la *intranet* de ARCOS:

- *https://arcos.inf.uc3m.es/stats_apache/arcos/*
- *https://arcos.inf.uc3m.es/stats_apache/jornadas/*
- *https://arcos.inf.uc3m.es/stats_apache/magsi/*
- *https://arcos.inf.uc3m.es/stats_apache/mimpi/*
- *https://arcos.inf.uc3m.es/stats_apache/xpn/*
- *https://arcos.inf.uc3m.es/stats_apache/winpfs/*

5.6.7. Instalación de un generador de estadísticas para el servidor SSH

Se ha creado un *script* que genera páginas web con las estadísticas de acceso del servidor *Caponata* (servidor SSH del sistema). Éste script utiliza el comando *freq* [3], una herramienta escrita en *perl* que analiza las entradas del fichero */var/log/lastlog* del sistema.

El script se localizará en */usr/local/bin/freq.sh* (ver contenido en página 302.), dentro de *Caponata*. Los pasos que realiza el *script* son los siguientes:

- Genera una primera página web con la salida del comando *freq*, utilizando una imagen como sustituto del asterisco, para crear un diagrama de barras. Éste ánalisis genera el número de accesos al servidor *Caponata* por cada usuario, y muestra una barra para crear una comparativa gráfica.
- Copia la web generada (*login_per_user.html*) al directorio *private_html/stats_user* del usuario *'web'*.
- Borra la web generada.
- Genera otra página web con la salida del comando *freq* solicitando las estadísticas anteriores pero con más detalle.
- Copia la web generada (*full.html*) al directorio *private_html/stats_user* del usuario *'web'*.

- Borra la web generada.
- Por último genera una web con la información dada por el comando *lastlog* (*lastlog.html*) y la copia al directorio *private_html/stats_user* del usuario *'web'*.
- Borra el último fichero generado.

El script debe ejecutarse todos los dias mediante el *crontab* de *Caponata*, por tanto se debe añadir la línea siguiente al *crontab* de *root*:

```
15 0 * * * /usr/local/bin/freq.sh
```

Las páginas *web* generadas se pueden consultar en la *intranet* de ARCOS, siendo los enlaces los siguientes:

- *https://arcos.inf.uc3m.es/stats_user/login_per_user.html*
- *https://arcos.inf.uc3m.es/stats_user/full.html*
- *https://arcos.inf.uc3m.es/stats_user/lastlog.html*

5.6.8. Instalación de PhpSysInfo

PhpSysInfo es una herramienta que genera una página web con información del sistema donde se ejecuta. Recopila esta información accediendo al *proc* del sistema. Se desea generar una web por cada una de las máquinas que componen el sistema, para ello, será necesario instalar el servidor Apache-ssl en cada una de las máquinas del sistema, aparte de *Piojito* (servidor web principal del sistema). Se instalará Apache-ssl configurado para permitir un acceso mínimo en el resto de máquinas (*Donald, Daisy, Piolin, Lucas* y *Caponata*). En *Boyerito* no se podrá instalar PhpSysInfo dado que estará apagada la mayor parte del tiempo (solo se enciende para realizar las copias de seguridad).

Por tanto, en todas las máquinas del sistema se instala el paquete *phpsysinfo*:

```
# apt-get install phpsysinfo
```

Después es necesario crear un enlace simbólico en la web segura que apunte al directorio */usr/share/phpsysinfo*, donde se genera la web con la información del sistema. En todas las máquinas donde se ha instalado Phpsysinfo, salvo en *Piojito*, se ejecuta:

```
# ln -s /usr/share/phpsysinfo /var/www-ssl/stats_sys
```

En *Piojito*, la web segura está alojada en el directorio *private_html* de la cuenta *'web'*, por tanto es necesario crear el enlace simbólico dentro de dicha cuenta:

```
# su - web
$ cd private_html
$ ln -s /usr/share/phpsysinfo stats_sys
$ exit
```

Los enlaces a las web generadas por PhpSysInfo aparecen en la *intranet* de ARCOS, y son los siguientes:

- **Estadísticas de *Donald***: *https://donald.arcos.inf.uc3m.es/stats_sys/*
- **Estadísticas de *Daisy***: *https://daisy.arcos.inf.uc3m.es/stats_sys/*
- **Estadísticas de *Piojito***: *https://arcos.inf.uc3m.es/stats_sys/*
- **Estadísticas de *Piolin***: *https://piolin.arcos.inf.uc3m.es/stats_sys/*
- **Estadísticas de *Lucas***: *https://lucas.arcos.inf.uc3m.es/stats_sys/*
- **Estadísticas de *Caponata***: *https://caponata.arcos.inf.uc3m.es/stats_sys/*

5.6.9. Instalación de Munin

Munin [6] es una herramienta de monitorización de máquinas que genera estadísticas sobre su funcionamiento. Utiliza las herramientas RRDTool para generar gráficas de rendimiento de los parámetros del sistema analizados. Utiliza una interfaz web para mostrar las gráficas generadas, y además es un software que permite trabajar de forma distribuida, mostrando la información de varias máquinas. Para ello se instala en una máquina la parte servidora de Munin y en el resto de máquinas la parte cliente, que mandará los datos recopilados al servidor para que éste los muestre.

La máquina donde se alojará el servidor Munin es *Piojito*, para ello se instalan los siguientes paquetes:

```
# apt-get install munin munin-node
```

Por defecto, la instalación genera el directorio */var/www/munin*, donde aloja los ficheros del interfaz web. En *Piojito*, no se utilizan los directorios */var/www* y */var/www-ssl* sino los directorios *public_html* y *private_html* respectivamente de la cuenta de usuario *'web'*. Se mueve el directorio */var/www/munin* a */var/lib/munin/html* y a continuación se genera un enlace simbólico dentro del directorio *private_html* de la cuenta *'web'* que apunte a la nueva localización:

```
# su - web
$ cd private_html
$ ln -s /var/lib/munin/html munin
$ exit
```

Se configura el fichero */etc/munin/munin.conf* para que obtenga la información del resto de máquinas donde estará instalada la parte cliente de Munin. Además en el fichero *munin.conf* (ver página 303.), se configura la nueva ruta donde alojar la información: */var/lib/munin/html*.

La parte servidor de Munin esta completamente configurada. Respecto a la parte cliente, hay que instalar el siguiente paquete en *Donald*, *Daisy*, *Piolin*, *Lucas* y *Caponata*:

```
#apt-get install munin-node
```

Y en todas las máquinas, incluyendo a *Piojito*, se configura el fichero */etc/munin/munin-node.conf* para permitir que el servidor de Munin (instalado en *Piojito*) tenga acceso al cliente Munin que se ejecuta en cada máquina. Como ejemplo, se adjunta en el anexo 4 del presente documento el fichero */etc/munin/munin-node.conf* de *Piojito* (ver página 304.).

Una vez configurados los clientes Munin y el servidor Munin, se inicia el cliente en todas las máquinas:

```
# /etc/init.d/munin-node start
```

El servidor de Munin se ejecuta automáticamente a través del *crontab* del sistema. El enlace existente en la *intranet* donde se accede al interfaz web de Munin es: *https://arcos.inf.uc3m.es/munin/*.

Capítulo 6

Resultados del nuevo sistema

6.1. Introducción

Una vez puesto en marcha el nuevo sistema de servidores, se comprueban los resultados obtenidos con los programas de monitorización instalados. Se obtendrán resultados de varios de los servicios instalados, así como información sobre el rendimiento de cada máquina. Con esta información se obtendrán conclusiones acerca del funcionamiento global del sistema que se utilizarán para comprobar que cumple con los objetivos propuestos, y en caso de necesidad, realizar las modificaciones oportunas. En las secciones siguientes se detallarán los resultados obtenidos.

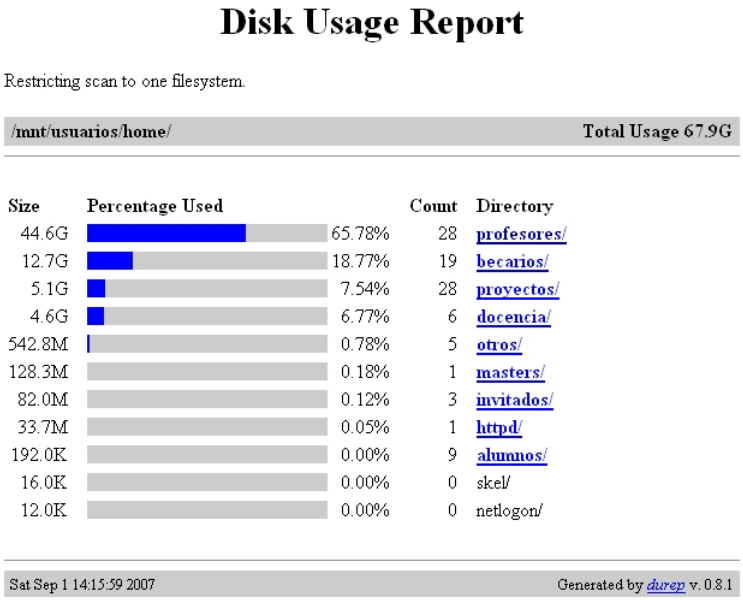
6.2. Servicio de almacenamiento

En esta sección se detalla el porcentaje de uso de cada uno de los directorios destinados a ser utilizados por los usuarios. Estos directorios son */mnt/home*, */mnt/mail*, */mnt/web* y */mnt/backup*.

6.2.1. /mnt/home

En la figura 6.1 se puede observar como el mayor porcentaje de utilización del directorio */mnt/home* lo tiene el grupo de profesores, con un 65,78 % respecto al total de datos de dicho directorio. Los becarios, por otro lado, tienen también un elevado porcentaje de utilización, muy seguidos por el grupo de proyectos y el grupo de docencia.

Estas estadísticas de uso reflejan el uso cotidiano que se hace del sistema, es decir, las personas que utilizan diariamente sus cuentas son los profesores y los becarios de ARCOS. El número de becarios es más reducido que el de profesores, por esa razón, del total de datos que contiene el directorio */mnt/home*, la mayor parte pertenece al grupo de profesores.



Disk Usage Report

Restricting scan to one filesystem.

/mnt/usuarios/home/ — Total Usage 67.9G

Size	Percentage Used	Count	Directory
44.6G	65.78%	28	profesores/
12.7G	18.77%	19	becarios/
5.1G	7.54%	28	proyectos/
4.6G	6.77%	6	docencia/
542.8M	0.78%	5	otros/
128.3M	0.18%	1	masters/
82.0M	0.12%	3	invitados/
33.7M	0.05%	1	httpd/
192.0K	0.00%	9	alumnos/
16.0K	0.00%	0	skel/
12.0K	0.00%	0	netlogon/

Sat Sep 1 14:15:59 2007 — Generated by *durep* v. 0.8.1

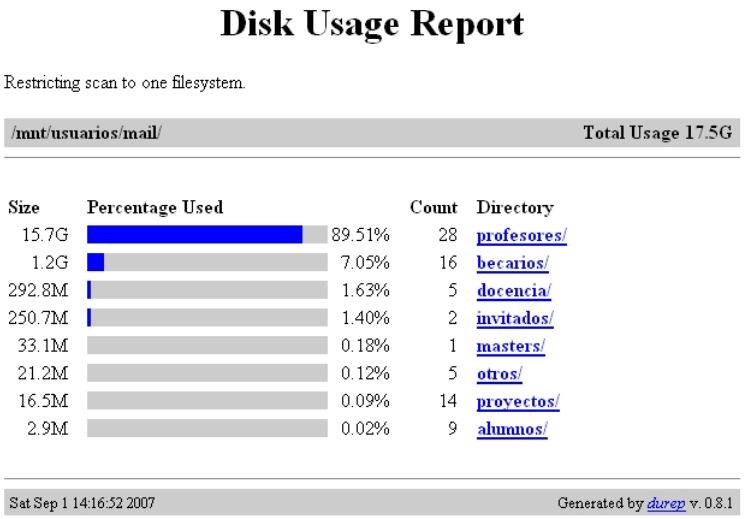
Figura 6.1: Ocupación del directorio */mnt/home*

6.2.2. /mnt/mail

En la figura 6.2 se puede observar algo similar al caso del directorio */mnt/home*. Las estadísticas reflejan que el grupo de profesores es el que hace un uso más intensivo del directorio */mnt/mail*. Esto es porque el personal de ARCOS, en su mayoría profesores, utilizan el correo electrónico diariamente, siendo una de las herramientas de comunicación más importantes. El grupo de becarios utiliza también el correo electrónico, pero el número de personas becarias es inferior al de profesores, hecho que se refleja en las estadísticas.

6.2.3. /mnt/web

Respecto al directorio */mnt/web*, como se ve en la figura 6.3, el grupo de profesores sigue siendo el que más utiliza este espacio de almacenamiento. Dado que este espacio está destinado a almacenar las páginas web de cada cuenta de usuario (la pública, *'public_html'*, y la privada *'private_html'*), se refleja en las estadísticas como el grupo de profesores cuenta con un mayor número de páginas web y datos almacenados en las mismas. No obstante, el grupo de cuentas de masters y docencia, utilizan también intensivamente éste directorio. Esto es debido a que para la docencia impartida en ARCOS, se crean múltiples páginas web, con información textual y también con ficheros de datos, para utilizar por parte de los profesores y los alumnos.



Disk Usage Report

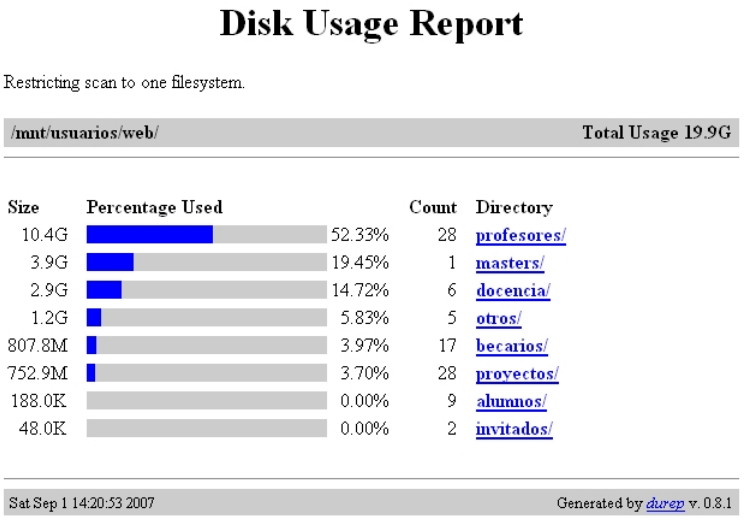
Restricting scan to one filesystem.

/mnt/usuarios/mail/ **Total Usage 17.5G**

Size	Percentage Used	Count	Directory
15.7G	89.51%	28	profesores/
1.2G	7.05%	16	becarios/
292.8M	1.63%	5	docencia/
250.7M	1.40%	2	invitados/
33.1M	0.18%	1	masters/
21.2M	0.12%	5	otros/
16.5M	0.09%	14	proyectos/
2.9M	0.02%	9	alumnos/

Sat Sep 1 14:16:52 2007 Generated by durep v. 0.8.1

Figura 6.2: Ocupación del directorio */mnt/mail*



Disk Usage Report

Restricting scan to one filesystem.

/mnt/usuarios/web/ **Total Usage 19.9G**

Size	Percentage Used	Count	Directory
10.4G	52.33%	28	profesores/
3.9G	19.45%	1	masters/
2.9G	14.72%	6	docencia/
1.2G	5.83%	5	otros/
807.8M	3.97%	17	becarios/
752.9M	3.70%	28	proyectos/
188.0K	0.00%	9	alumnos/
48.0K	0.00%	2	invitados/

Sat Sep 1 14:20:53 2007 Generated by durep v. 0.8.1

Figura 6.3: Ocupación del directorio */mnt/web*

6.2.4. /mnt/backup

El último de los directorios dedicados al almacenamiento para las cuentas de usuario del sistema, es decir, */mnt/backup*, no esta dividido por grupos, sino directamente por el nombre de las cuentas. En la figura 6.4, se puede observar como las cuentas *folcina* y *acaldero* son las que más utilizan este espacio. Estas cuentas pertenecen a los administradores, que por conocer las ventajas del sistema, hacen un uso más óptimo del mismo. El resto de cuentas que aparecen, son de las personas que han utilizado este espacio de almacenamiento.

Una de las conclusiones que se obtienen de ésta gráfica, es la poca utilización de este directorio por parte del personal de ARCOS. Una de las razones puede ser el desconocimiento del mismo, hecho que se solventaría publicitando éste espacio (por correo electrónico, por ejemplo); por otro lado, otra de las razones puede ser la preferencia de los usuarios a utilizar únicamente su directorio *home* para almacenar todos sus datos.

Disk Usage Report

Restricting scan to one filesystem.

/mnt/backup_usuarios/ **Total Usage 52.3G**

Size	Percentage Used	Count	Directory
22.1G	42.36%	5	folcina/
8.8G	16.91%	3	acaldero/
8.5G	16.26%	2	jcarrete/
3.6G	6.98%	2	arodriguez/
2.9G	5.54%	1	jdaniel/
1.9G	3.72%	1	lprada/
1.4G	2.65%	4	fgarcia/
1.1G	2.13%	0	desingh/
1.1G	2.08%	0	web/
360.7M	0.67%	1	mreyes/
215.2M	0.40%	4	jfernand/
88.6M	0.17%	2	libros/
41.9M	0.08%	0	bbergua/
20.7M	0.04%	0	cupcam/
1.3M	0.00%	0	stf/
4.0K	0.00%	1	[FILES]

Sat Sep 1 14:21:33 2007 Generated by *durep* v. 0.8.1

Figura 6.4: Ocupación del directorio */mnt/backup*

6.3. Servicio de resolución de nombres

El servicio de resolución de nombres (DNS) está instalado en la máquina virtual *Piojito*. La figura 6.5 refleja las estadísticas de uso del DNS de ARCOS. Como se puede observar, la mayoría de las peticiones al DNS son de tipo *A*, seguidas por las del tipo *MX*.

Las consultas de tipo *A*, solicitan la IP asociada al nombre de la máquina o dominio especificado, y generalmente se producen por la navegación a través de internet. Cuando una máquina tiene como servidor DNS, el servidor de ARCOS (163.117.148.240), y el usuario navega por internet, constantemente está introduciendo direcciones de dominios, que el servidor DNS resuelve para que el navegador conozca la IP. Las consultas de tipo

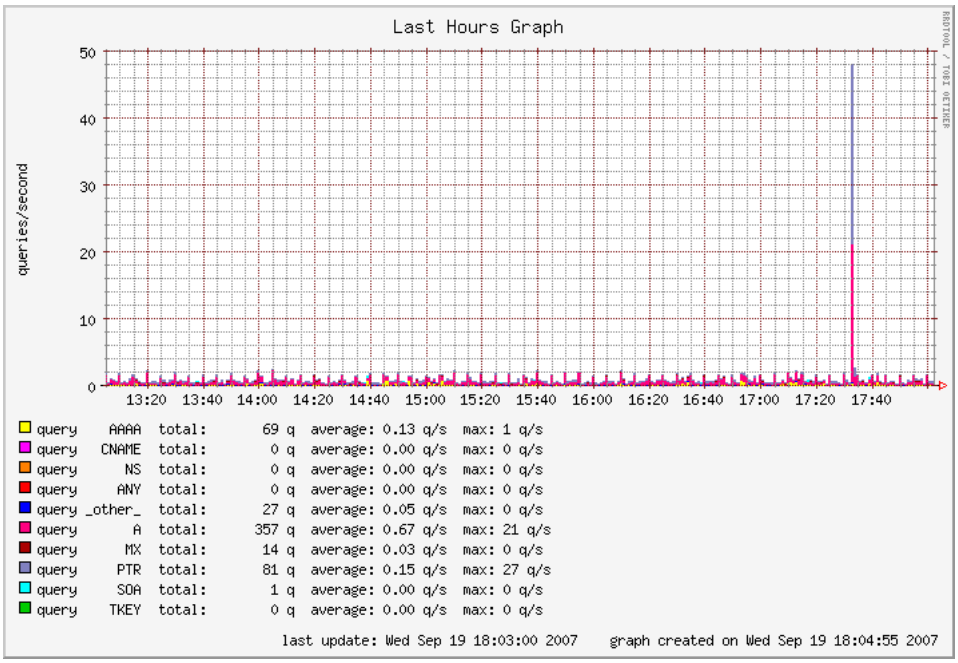


Figura 6.5: Estadísticas de utilización del DNS de ARCOS

MX, solicitan la IP asociada al servidor de correo (o servidores) del dominio especificado. Este tipo de consultas se producen cuando se envían correos electrónicos utilizando el SMTP de ARCOS. Según se ve en la figura 6.5, se han producido 14 consultas de tipo *MX*, lo que se traduce en el envío de correos a 14 servidores distintos.

Otra de los puntos a destacar, es el pico que aparece entre las 17:20 y las 17:40, con consultas de tipo *A* y *PTR* (consultas inversas). Si se revisa el *log* del servicio DNS entre esas horas, se ven muchas consultas seguidas de este tipo:

```
Sep 19 17:33:13.203 client 163.117.148.245#33301: query: pupa.it.uc3m.es IN A
Sep 19 17:33:13.211 client 163.117.148.245#33301: query: 91.140.117.163.in-addr.arpa IN PTR
```

Estas consultas se pueden interpretar como un ataque de DNS *Spoofing*; el atacante envía al servidor DNS un número elevado de consultas por segundo con la IP de origen falseada.

Lo que pretende el ataque es que el servidor DNS colapse la IP de origen que ha falseado, es decir, pretende que *Piojito* colapse a *Caponata* con las respuestas de las peticiones realizadas.

6.4. Servicio de correo

Los resultados del sistema dados por el servicio de correo, se dividirán en 3 apartados: por un lado se mostrarán las estadísticas de Postfix, por otro lado las estadísticas de conexión a los servidores Pop3 e Imap, y por último, las estadísticas del software antivirus Amavis.

6.4.1. Postfix

En la figura 6.6 aparece dividida en dos secciones. La parte de arriba refleja el número de correos recibidos y enviados por Postfix; la parte de abajo refleja la cantidad de dichos correos que son invalidados o marcados como virus o *spam*. Se puede observar como se

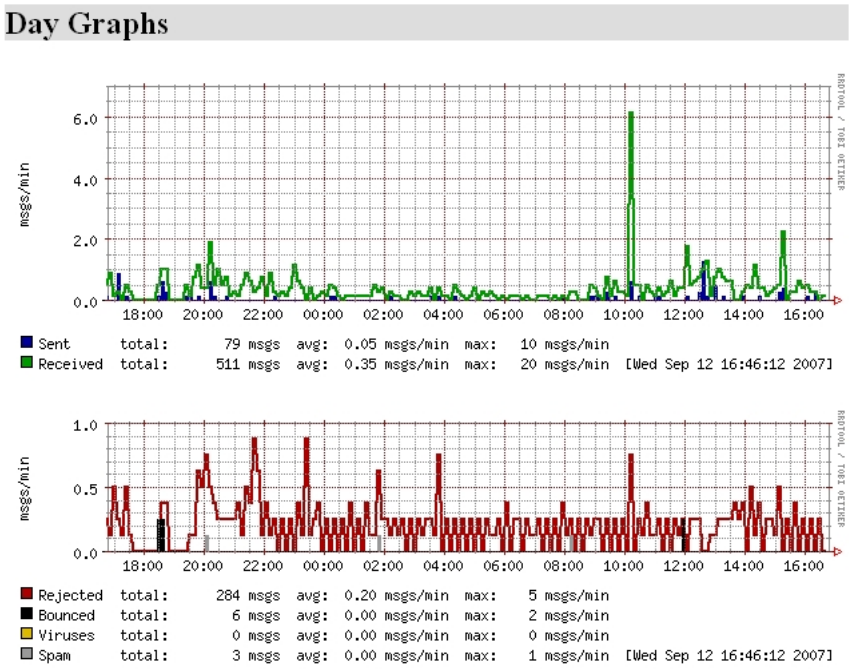


Figura 6.6: Estadísticas de utilización del servidor Postfix

han recibido un total de 511 mensajes, y otros 284 han sido rechazados. Los rechazos se producen por la cantidad de correo '*basura*' que entra en el sistema y es detectado por los filtros utilizados en Postfix.

La mayor parte de los correos se reciben de día, entre las 8:00 y las 23:00. Esto también sucede en los envíos, pero en un horario más reducido: la mayor parte de correos se envían entre las 9:00 y las 21:00, las horas en las que el personal de ARCOS trabaja en la Universidad. Los pocos correos enviados que aparecen entre las 22:00 y las 8:00, corresponden a los *emails* con los informes de las copias de seguridad, enviados a los administradores.

6.4.2. Pop3 e Imap

La figura 6.7 esta dividida en dos secciones: la sección de arriba refleja las conexiones realizadas a los servidores Pop3 e Imap de ARCOS, y la sección de abajo refleja las conexiones realizadas a los servidores Pop3-SSL e Imap-SSL de ARCOS. Uno de los datos

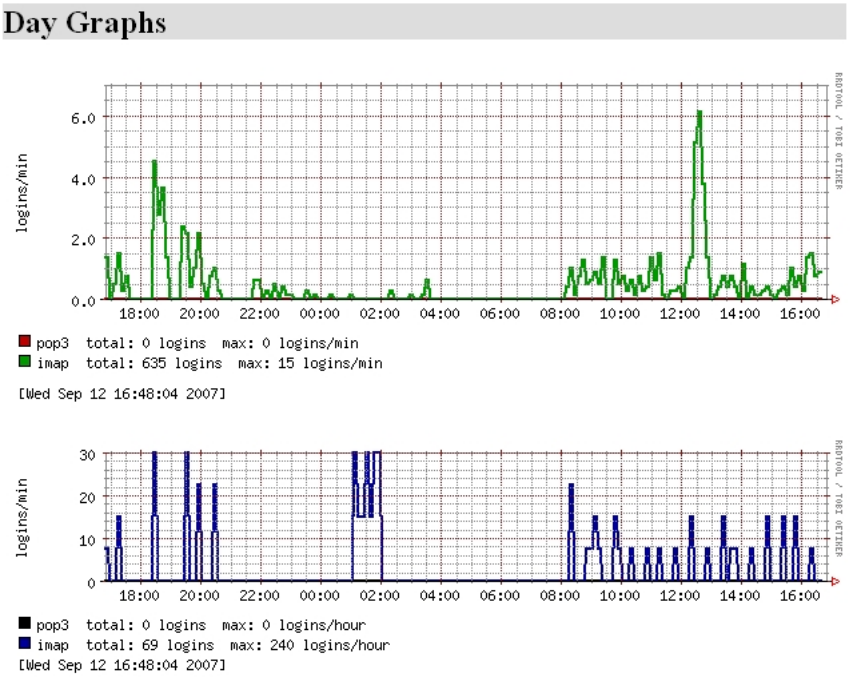


Figura 6.7: Estadísticas de utilización de los servidores Pop3 e Imap

más significativos de la figura 6.7 es la no utilización de los servidores Pop3 y Pop3-SSL. El personal de ARCOS solo utiliza Imap e Imap-SSL para utilizar su cuenta de correo electrónico. Se puede observar también como la mayor parte de las conexiones se realizan por la mañana, desde las 8:00 horas, y van decreciendo hasta algo más de las 18:00. Las conexiones nocturnas, desde las 22:00 hasta las 01:00, pueden corresponder a las personas

que leen su correo desde casa, antes de acostarse.

Las conexiones Imap-SSL reflejan de manera más evidente, el uso horario de acceso al correo. Dado que aparecen menos conexiones, se concluye que son pocas las personas que utilizan este tipo de conexión, o incluso solo una persona. Estas conexiones se producen en horario de trabajo, desde las 9:00 hasta las 18:00, probablemente realizadas desde la Universidad, y después en horario nocturno, desde las 21:00 hasta las 01:00, probablemente realizadas desde casa.

6.4.3. Amavis

La figura 6.8 esta dividida en dos secciones: la sección de arriba muestra el número de mensajes escaneados por Amavis y cuáles de éstos son baneados o marcados como infectados; la sección de abajo muestra el número de virus detectados en los correos escaneados. Se puede observar como Amavis ha escaneado todos los correos que han entrado

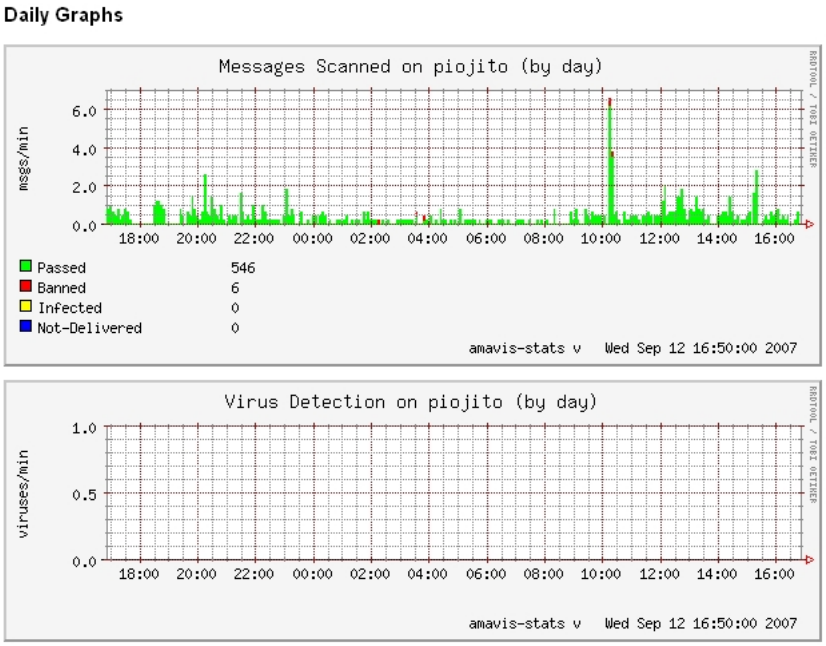


Figura 6.8: Estadísticas de Amavis

en el sistema en las últimas 24 horas y sólo 6 han sido baneados, pero no por contener virus. El hecho de que no se detecten virus puede ser debido a que todos los correos que llegan al SMTP de ARCOS, pasan previamente por el SMTP de la Universidad, que posee filtros para eliminar aquellos correos que contengan virus. Por ese motivo, es poco probable que entren correos desde fuera de la Universidad con virus. Los casos más pro-

bables de detección, serían de correos enviados dentro del segmento de ARCOS, pero los ordenadores utilizados en los despachos y laboratorios cuentan con antivirus que reducen la probabilidad de que esto suceda.

6.5. Servicio web

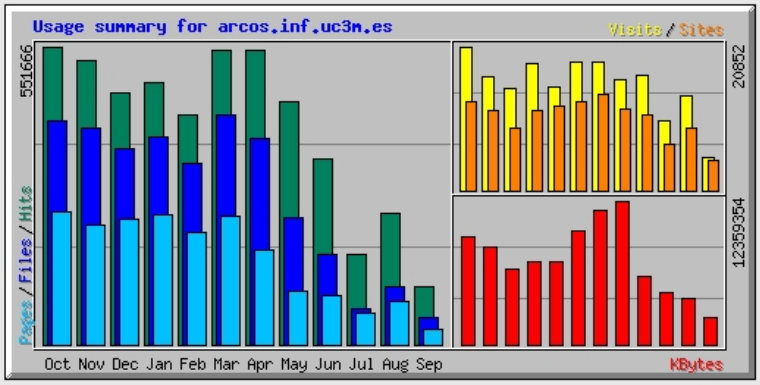
De los resultados ofrecidos por el servidor Web de ARCOS, instalado en *Piojito*, se analizará solo el dominio principal. Este domino es *arcos.inf.uc3m.es* (o *www.arcos.inf.uc3m.es*), que es el más significativo de ARCOS, por delante de *jornadas.arcos.inf.uc3m.es*, *magsi.arcos.inf.uc3m.es*, *mimpi.arcos.inf.uc3m.es*, *xpn.arcos.inf.uc3m.es*, y *winpfs.arcos.inf.uc3m.es*.

6.5.1. Dominio principal de ARCOS (*www.arcos.inf.uc3m.es*)

La figura 6.9 refleja los resultado obtenidos con el software Webalizer, donde se puede constatar que el uso del dominio *arcos.inf.uc3m.es*, esta claramente influenciado por el curso académico.

Los meses de más uso van desde octubre hasta junio, siendo los meses de verano los que menos visitas tienen. Esto refleja también que los meses de más trabajo y actividad del grupo ARCOS son los del curso académico, siendo la razón principal el uso de la web para la docencia. En la gráfica se observa que el mes de octubre, en el que comienza el curso académico es el mes que más visitas tiene, y va disminuyendo según se acerca febrero, el mes en el que se realizan los exámenes. Después de los exámenes hay un aumento notable de visitas los dos siguientes meses, decayendo el número de las mismas según se acerca la fecha de los exámenes del segundo cuatrimestre en junio.

La conclusión que se obtiene de estos datos es que la web del dominio principal de ARCOS es usada en su mayor parte por cuestiones docentes.



Summary by Month

Month	Daily Avg				Monthly Totals					
	Hits	Files	Pages	Visits	Sites	KBytes	Visits	Pages	Files	Hits
Sep 2007	8966	4223	2437	392	4333	2387166	4706	29245	50680	107594
Aug 2007	7832	3453	2560	439	8956	3990834	13622	79365	107069	242805
Jul 2007	5385	2142	1848	327	6673	4502018	10155	57312	66411	166948
Jun 2007	11437	5584	3030	559	11022	5901964	16788	90910	167523	343133
May 2007	14519	7559	3216	521	11830	12359354	16164	99711	234353	450109
Apr 2007	18190	12725	5806	618	13959	11551524	18546	174195	381754	545716
Mar 2007	17581	13749	7698	601	12950	9772165	18656	238644	426235	545024
Feb 2007	15181	12024	7436	538	12273	7072231	15070	208223	336685	425081
Jan 2007	15669	12422	7755	596	11562	7103059	18494	240432	385103	485743
Dec 2006	15062	11739	7493	476	9075	6461394	14786	232307	363932	466926
Nov 2006	17522	13319	7417	546	11679	8368223	16404	222525	399570	525664
Oct 2006	17795	13325	7936	672	12893	9224610	20852	246017	413086	551666
Totals						**88694542**	**184243**	**1918886**	**3332401**	**4856409**

Figura 6.9: Estadísticas de la página principal de ARCOS

6.6. Servicio de gestión de bases de datos con MySQL

En la figura 6.10 se puede observar el uso diario del servidor MySQL del sistema, instalado en *Piojito*. EL servicio MySQL se utiliza principalmente para dar soporte a la página web del grupo, que obtiene su información de la base de datos.

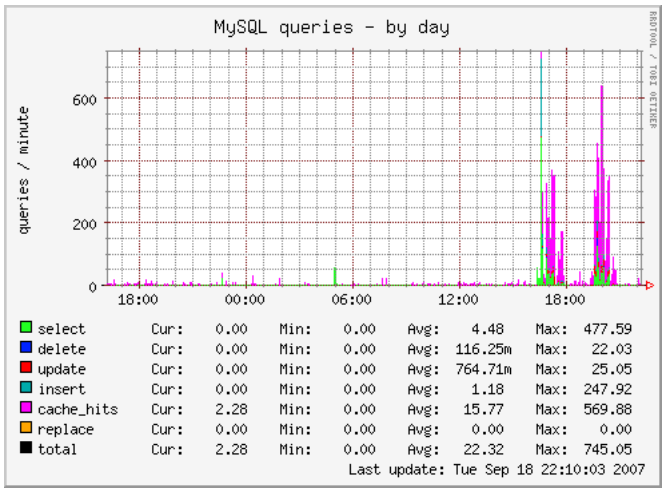


Figura 6.10: Consultas diarias en el servidor MySQL de *Piojito*

Este hecho se refleja en la gráfica, dado que la mayoría de las peticiones al servidor MySQL son de tipo *cache_hits*. La explicación es la siguiente: cuando una persona solicita la página web de ARCOS en su navegador, se produce una select al servidor MySQL. Cuando esa persona u otras, en un periodo reducido de tiempo, solicita la misma página, el servidor MySQL devuelve los datos que mantiene en cache, respecto a esa *select*, dado que la información de esa base de datos no se ha modificado. En la gráfica se comprueba como antes de realizar peticiones de tipo *cache_hits*, el servidor MySQL realiza peticiones de tipo *select*.

También es importante destacar que aparte de la página web de ARCOS, hay un Moodle instalado y en periodo de pruebas. La herramienta Moodle, se utiliza para crear portales de docencia virtuales, mediante páginas web. Además utiliza el servidor MySQL para alojar toda la información, por lo tanto, pueden producirse peticiones de todo tipo, para actualizar la base de datos.

De la gráfica se concluye que el servidor MySQL da un rendimiento óptimo al alcanzar picos de respuesta de hasta 570 peticiones por segundo, y que la mayor parte del uso se produce dentro del horario de trabajo: entre las 08:00 horas y las 20:00 horas (la captura muestra para ese día una utilización más notable en horas de tarde, alrededor de las 17:00 y las 20:00 horas).

6.7. Servicio ssh

La figura 6.11 refleja los datos del lastlog de *Caponata*. Esta salida la produce el *script freq.sh* en *Caponata* (ver página 8.4.40.)

Se puede observar como todos los días se utiliza éste servidor como máquina SSH para que el personal de ARCOS tenga acceso remoto a una máquina Linux. La mayoría de las conexiones se producen en horario de trabajo y casi siempre se conecta el usuario *root* (por parte de algún administrador), para las tareas de monitorización y revisión de *logs*, dado que *Caponata* es una máquina con acceso a datos muy críticos.

```
Day of Wk Logins   Graph
========= ======== ========================
Mon       9        ############################
Tue       6        ##################
Wed       2        #####


Day in Mo Logins   Graph
========= ======== ========================
10        9        ############################
11        6        ##################
12        2        #####


Login Hr. Logins   Graph
========= ======== ========================
00        2        #############
09        1        ######
12        1        ######
13        1        ######
15        2        #############
16        4        ############################
17        2        #############
19        3        ###################
22        1        ######


Name      Logins   Graph
========= ======== ========================
acaldero  6        ############################
fgarcia   1        ####
folcina   3        #############
lprada    2        ########
root      5        ######################


Month     Logins   Graph
========= ======== ========================
Sep       17       ############################


Terminal  Logins   Graph
========= ======== ========================
pts/0     13       ############################
pts/1     4        #######
```

Figura 6.11: Estadísticas de acceso en *Caponata* con detalles

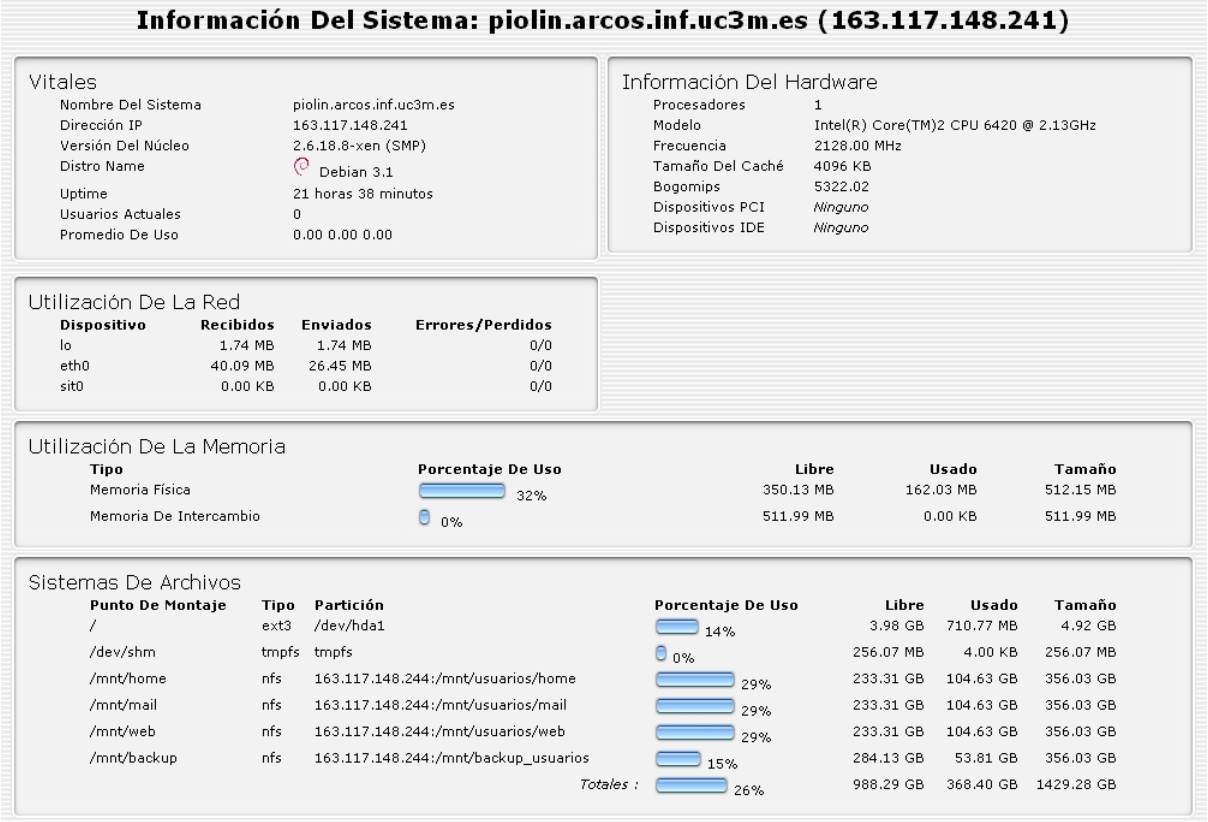
6.8. Información del sistema de *Piolin*

Piolin es la máquina virtual dedicada a la autenticación. En la página 94. se muestran todas las características del diseño de ésta máquina. En los siguientes apartados se verán los resultados obtenidos por ella.

6.8.1. Información general

En la figura 6.12 se ve la gráfica obtenida por la herramienta Phpsysinfo. Esta figura se divide en varios apartados que se tratarán a continuación:

- **Vitales**: en éste apartado se puede comprobar la versión del *kernel* que está ejecutando la máquina (*2.6.18.8-xen*), que se trata de una distribución Debian, el tiempo que lleva arrancada desde el último reinicio, y la IP (163.117.148.241), entre otros datos.
- **Información del hardware**: la información que muestra es la relativa a la máquina física donde se ejecuta *Piolin*, siendo en este caso *Donald* (Intel Core 2 Duo).
- **Utilización de la red**: muestra información relativa al uso de cada interfaz de red (*lo* y *eth0*).
- **Utilización de la memoria**: de los 512Mb de memoria RAM asignados a *Piolin*, se comprueba como ésta utiliza un 32 %, no teniendo que utilizar la memoria *swap*. Por lo tanto, 512Mb son óptimos para ésta máquina virtual.
- **Sistemas de archivos**: los sistemas de ficheros propios de *Piolin* son el raíz y el sistema virtual */dev/shm*. El sistema raíz de *Piolin* está ocupado en un 14 %, por lo tanto no se prevee ningún tipo de incidencia respecto al espacio necesitado.



Información Del Sistema: piolin.arcos.inf.uc3m.es (163.117.148.241)

Vitales

Nombre Del Sistema	piolin.arcos.inf.uc3m.es
Dirección IP	163.117.148.241
Versión Del Núcleo	2.6.18.8-xen (SMP)
Distro Name	Debian 3.1
Uptime	21 horas 38 minutos
Usuarios Actuales	0
Promedio De Uso	0.00 0.00 0.00

Información Del Hardware

Procesadores	1
Modelo	Intel(R) Core(TM)2 CPU 6420 @ 2.13GHz
Frecuencia	2128.00 MHz
Tamaño Del Caché	4096 KB
Bogomips	5322.02
Dispositivos PCI	*Ninguno*
Dispositivos IDE	*Ninguno*

Utilización De La Red

Dispositivo	Recibidos	Enviados	Errores/Perdidos
lo	1.74 MB	1.74 MB	0/0
eth0	40.09 MB	26.45 MB	0/0
sit0	0.00 KB	0.00 KB	0/0

Utilización De La Memoria

Tipo	Porcentaje De Uso	Libre	Usado	Tamaño
Memoria Física	32%	350.13 MB	162.03 MB	512.15 MB
Memoria De Intercambio	0%	511.99 MB	0.00 KB	511.99 MB

Sistemas De Archivos

Punto De Montaje	Tipo	Partición	Porcentaje De Uso	Libre	Usado	Tamaño
/	ext3	/dev/hda1	14%	3.98 GB	710.77 MB	4.92 GB
/dev/shm	tmpfs	tmpfs	0%	256.07 MB	4.00 KB	256.07 MB
/mnt/home	nfs	163.117.148.244:/mnt/usuarios/home	29%	233.31 GB	104.63 GB	356.03 GB
/mnt/mail	nfs	163.117.148.244:/mnt/usuarios/mail	29%	233.31 GB	104.63 GB	356.03 GB
/mnt/web	nfs	163.117.148.244:/mnt/usuarios/web	29%	233.31 GB	104.63 GB	356.03 GB
/mnt/backup	nfs	163.117.148.244:/mnt/backup_usuarios	15%	284.13 GB	53.81 GB	356.03 GB
		Totales :	26%	988.29 GB	368.40 GB	1429.28 GB

Figura 6.12: Información general sobre *Piolin*

6.8.2. Tráfico de red

La figura 6.13 muestra el tráfico de entrada y salida del interfaz eth0 de *Piolin*. Hay dos picos en la gráfica que siempre van a ser reconocibles: el que sucede a las 02:00 horas, y el de las 5:51 horas. Estos picos corresponden a los *backups* programados del raíz de *Piolin* y del servicio Ldap. El resto de picos que aparecen en la gráfica son provocados por múltiples peticiones en ese instante al servidor Ldap o por operaciones realizadas en *Piolin* sobre una cuenta de usuario.

Concretamente, el pico que aparece sobre las 14:40 horas, fue provocado por una operación de tipo *chown* y *chmod* recursiva sobre una cuenta de usuario. El cliente NFS en esos momentos recibió mucha información sobre la estructura de directorios de dicha cuenta, y devolvía las operaciones a realizar sobre ese sistema. Esto se comprueba más adelante en la gráfica sobre el cliente NFS de *Piolin* (ver siguiente apartado).

Se concluye que el sistema *Piolin* genera un tráfico leve, provocado por los *backups* realiza-

dos, por las peticiones al servidor Ldap, y por las operaciones de administración realizadas en la máquina.

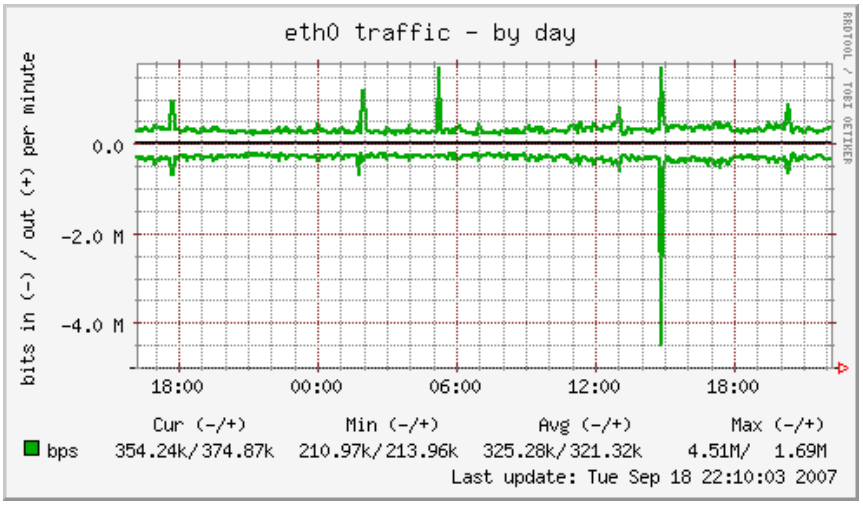


Figura 6.13: Trafico diario por el interfaz *eth0* en *Piolin*

6.8.3. Cliente NFS

En la figura 6.14 se comprueba la utilización del cliente NFS en la máquina *Piolin*. Únicamente aparece un pico sobre las 14:40; esto es debido a la operación realizada que se describía en el apartado anterior. Esta operación realiza lecturas de atributos de ficheros sobre los sistemas de ficheros importados en *Piolin*.

Se concluye que el tráfico del cliente NFS en *Piolin* solo se produce al realizar operaciones de administración de cuentas de usuarios en la propia máquina virtual.

6.8.4. Uso de CPU

La figura 6.15 muestra el uso de CPU diario de *Piolin*. Se puede comprobar como la mayor parte del tiempo la CPU está desocupada. Esto es porque los únicos servicios que se utilizan de *Piolin*, son el servidor de Ldap y el servidor de Samba (para autenticación). Ambos demonios utilizan muy pocos recursos de la máquina, hecho que se refleja en la gráfica.

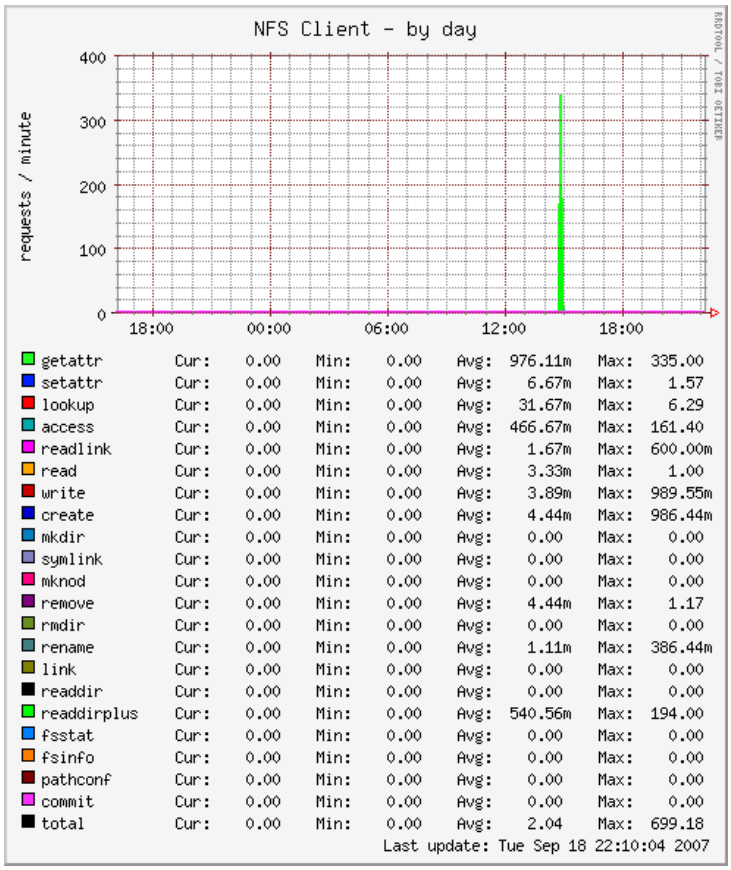


Figura 6.14: Cliente NFS diario en *Piolin*

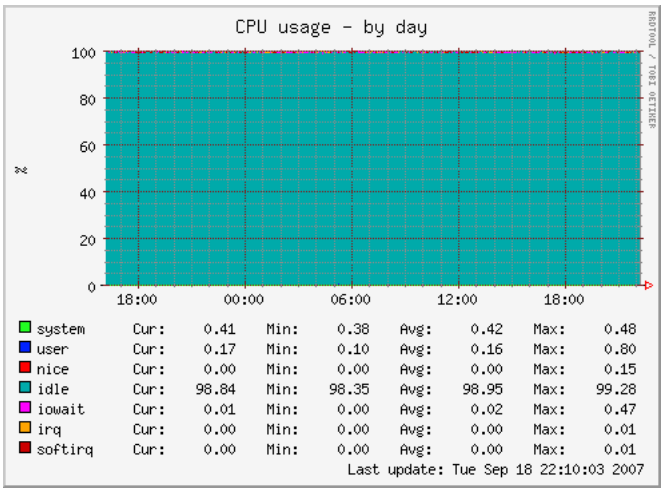


Figura 6.15: Uso de CPU diario en *Piolin*

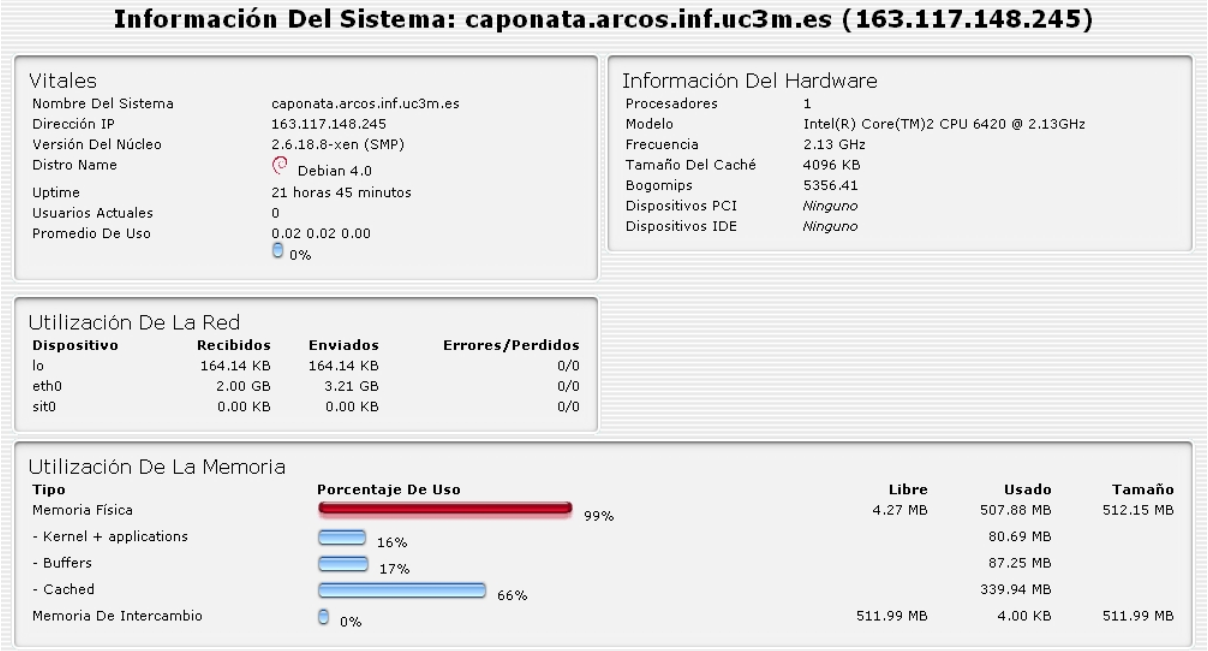
6.9. Información del sistema de *Caponata*

Caponata es la máquina virtual destinada a servir como terminal remoto SSH. En la página 112. se muestran todas las características del diseño de ésta máquina. En los siguientes apartados se verán los resultados obtenidos por ella.

6.9.1. Información general

En las figuras 6.16 y 6.17, se ven las gráficas obtenidas por la herramienta Phpsysinfo. Esta figura se divide en varios apartados que se tratarán a continuación:

- **Vitales**: en éste apartado se puede comprobar la versión del *kernel* que está ejecutando la máquina (*2.6.18.8-xen*), que se trata de una distribución Debian, el tiempo que lleva arrancada desde el último reinicio, y la IP (163.117.148.245), entre otros datos.
- **Información del hardware**: la información que muestra es la relativa a la máquina física donde se ejecuta *Caponata*, siendo en este caso *Donald* (Intel Core 2 Duo).
- **Utilización de la red**: muestra información relativa al uso de cada interfaz de red (*lo* y *eth0*).
- **Utilización de la memoria**: de los 512Mb de memoria RAM asignados a *Caponata*, se comprueba como ésta utiliza un 99 %, obligando a utilizar la memoria *swap*, de la que utiliza un 66 %. Por lo tanto, es recomendable aumentar la memoria RAM de *Caponata* en otros 512Mb, aunque *Donald* no tiene suficiente memoria para ello. En un futuro se ampliará la memoria de *Donald* para poder realizar esta modificación en *Caponata*.
- **Sistemas de archivos**: los sistemas de ficheros propios de *Piolin* son el raíz y los sistemas virtuales de tipo *tmpfs*. El sistema raíz de *Caponata* está ocupado en un 21 %, por lo tanto no se prevee ningún tipo de incidencia respecto al espacio necesitado.



Información Del Sistema: caponata.arcos.inf.uc3m.es (163.117.148.245)

Vitales

Nombre Del Sistema	caponata.arcos.inf.uc3m.es
Dirección IP	163.117.148.245
Versión Del Núcleo	2.6.18.8-xen (SMP)
Distro Name	Debian 4.0
Uptime	21 horas 45 minutos
Usuarios Actuales	0
Promedio De Uso	0.02 0.02 0.00
	0%

Información Del Hardware

Procesadores	1
Modelo	Intel(R) Core(TM)2 CPU 6420 @ 2.13GHz
Frecuencia	2.13 GHz
Tamaño Del Caché	4096 KB
Bogomips	5356.41
Dispositivos PCI	*Ninguno*
Dispositivos IDE	*Ninguno*

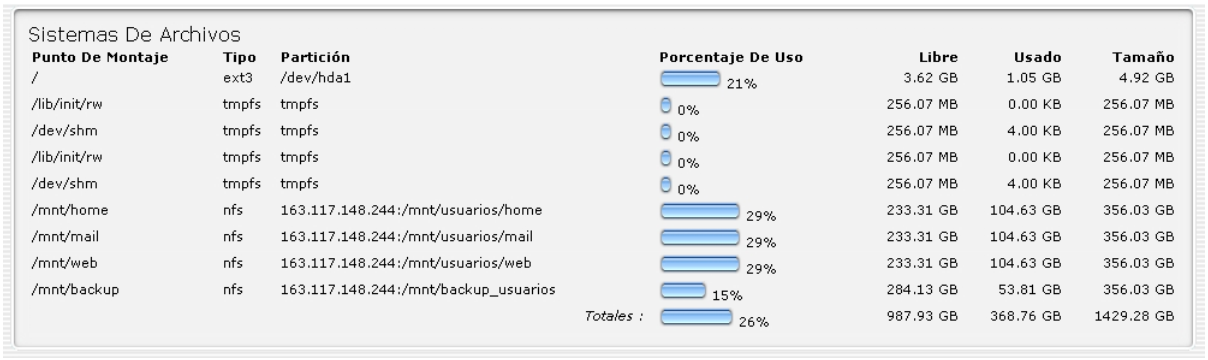
Utilización De La Red

Dispositivo	Recibidos	Enviados	Errores/Perdidos
lo	164.14 KB	164.14 KB	0/0
eth0	2.00 GB	3.21 GB	0/0
sit0	0.00 KB	0.00 KB	0/0

Utilización De La Memoria

Tipo	Porcentaje De Uso	Libre	Usado	Tamaño
Memoria Física	99%	4.27 MB	507.88 MB	512.15 MB
- Kernel + applications	16%		80.69 MB	
- Buffers	17%		87.25 MB	
- Cached	66%		339.94 MB	
Memoria De Intercambio	0%	511.99 MB	4.00 KB	511.99 MB

Figura 6.16: Información general sobre *Caponata* (1ª parte)



Sistemas De Archivos

Punto De Montaje	Tipo	Partición	Porcentaje De Uso	Libre	Usado	Tamaño
/	ext3	/dev/hda1	21%	3.62 GB	1.05 GB	4.92 GB
/lib/init/rw	tmpfs	tmpfs	0%	256.07 MB	0.00 KB	256.07 MB
/dev/shm	tmpfs	tmpfs	0%	256.07 MB	4.00 KB	256.07 MB
/lib/init/rw	tmpfs	tmpfs	0%	256.07 MB	0.00 KB	256.07 MB
/dev/shm	tmpfs	tmpfs	0%	256.07 MB	4.00 KB	256.07 MB
/mnt/home	nfs	163.117.148.244:/mnt/usuarios/home	29%	233.31 GB	104.63 GB	356.03 GB
/mnt/mail	nfs	163.117.148.244:/mnt/usuarios/mail	29%	233.31 GB	104.63 GB	356.03 GB
/mnt/web	nfs	163.117.148.244:/mnt/usuarios/web	29%	233.31 GB	104.63 GB	356.03 GB
/mnt/backup	nfs	163.117.148.244:/mnt/backup_usuarios	15%	284.13 GB	53.81 GB	356.03 GB
		Totales :	26%	987.93 GB	368.76 GB	1429.28 GB

Figura 6.17: Información general sobre *Caponata* (2ª parte)

6.9.2. Conexiones realizadas y tráfico del interfaz *eth0*

En la figura 6.18 se pueden observar las conexiones de red realizadas a *Caponata*. Es importante destacar la periodicidad de los picos que aparecen en la gráfica, salvo en el tramo horario de 12:00 a 18:00 horas aproximadamente. Esta periodicidad en las conexiones se produce por los servicios de monitorización, que abren conexiones periódicamente en *Caponata*, para recopilar información sobre la misma (como por ejemplo la herramienta Munin).

La curva irregular que aparece de las 12:00 a las 18:00 horas se produce por las conexiones de los usuarios, que probablemente utilizaron esas horas para conectarse a *Caponata*, el día que se realizó la captura de la gráfica.

Otras irregularidades que pueden aparecer, son los intentos de conexión fallidos por parte de atacantes. Éstas personas, establecen conexiones *tcp* con *Caponata* a través de SSH o Telnet a otros puertos, por lo tanto aparecerían en la gráfica como conexiones establecidas, generando picos muy altos si se trata de un ataque con muchas conexiones (como por ejemplo un ataque por fuerza bruta para entrar por SSH). Respeto al tráfico del

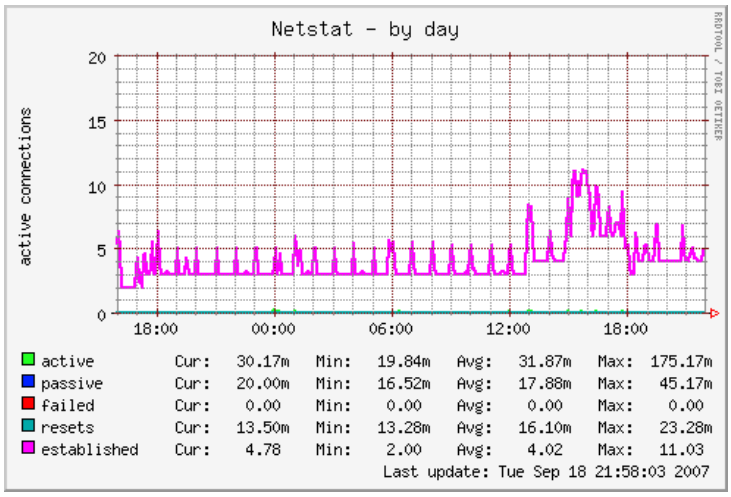


Figura 6.18: Conexiones diarias realizadas en *Caponata*

interfaz *eth0* (ver figura 6.19), lo más destacable son los picos periódicos que se producen entre las 06:00 y las 18:00 horas, correspondientes al envío de información para la monitorización de Snort. Estas transferencias se comentarán en el apartado de *Caponata* como cliente NFS, en la página 187. También es destacable el pico que aparece entorno a las 21:00 horas, debido a la escritura de datos en una cuenta de usuario. También se comentará en el apartado de *Caponata* como cliente NFS.

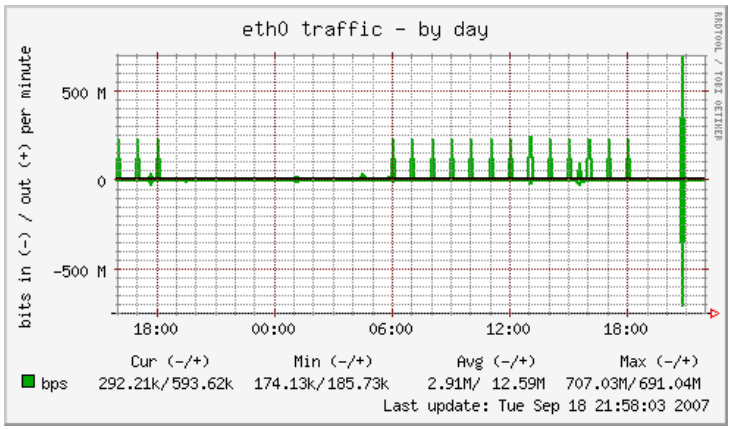


Figura 6.19: Tráfico diario del interfaz *eth0* en *Caponata*

6.9.3. Cliente NFS

La figura 6.20 muestra una gráfica con las estadísticas de uso del cliente NFS instalado en *Caponata*. Se puede observar una periodicidad de picos entre las 06:00 y las 18:00 horas. Esto corresponde a la ejecución de un *script* en *Piojito* que genera estadísticas que almacena en el directorio *private_html* de la cuenta *web*, a través de *Caponata*. El fragmento del *crontab* de *root* de la máquina *Piojito* que aparece a continuación, muestra las horas a las que se ejecuta el *script*:

```
#
# generate SnortSnarf statistics every hour from 6am to 6pm
#
0 6,7,8,9,10,11,12,13,14,15,16,17,18 * * * /usr/local/bin/snortsnarf.sh
```

Dentro del *script* hay una línea que es la que copia los datos generados a la *intranet* de ARCOS, utilizando una conexión a *Caponata*:

```
scp -r /tmp/stats_snort/  web@ssh:~/private_html/
```

Por otro lado, aparece un pico a las 19:00 que se corresponderá a la conexión de algún usuario que realice operaciones en su cuenta.

Al final de la gráfica, sobre las 21:00 horas, hay un pico rojo oscuro. Éste último pico se corresponde a una escritura en el sistema de ficheros importado, es decir, algún usuario ha escrito datos en uno de sus directorios (*/mnt/home*, */mnt/mail*, */mnt/web* o */mnt/backup*). Según muestran los datos de la gráfica, éste último pico (de tipo *write*), alcanza unos 2,66k de peticiones por minuto. Se han realizado 2.660 peticiones de escritura que seguramente se refieran a la copia de 2.660 ficheros distintos.

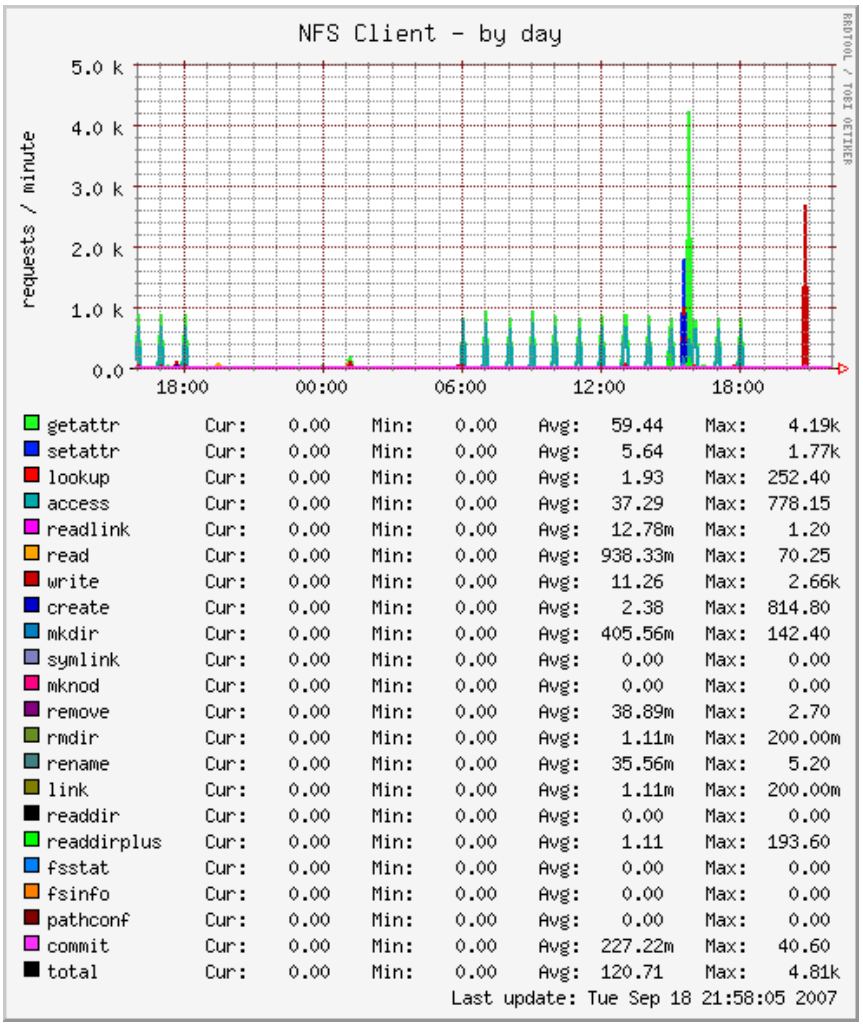


Figura 6.20: Cliente NFS diario en *Caponata*

6.9.4. Uso de CPU

En la figura 6.21 se puede observar el uso de CPU diario de la máquina *Caponata*. Las estadísticas reflejan que la mayor parte del tiempo, la CPU esta desocupada, con una media del 97.58 % en las últimas 24 horas respecto al dia de la captura. El resto del tiempo de ocupación de la CPU se divide, por orden, en los siguientes elementos: el tiempo del sistema, el tiempo del usuario y el tiempo de espera para las peticiones de entrada y salida.

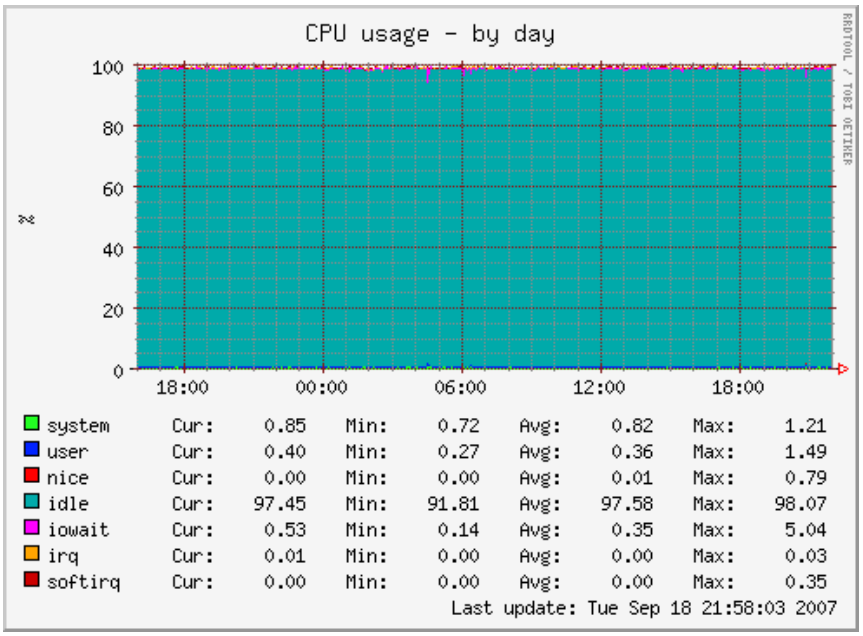


Figura 6.21: Uso de CPU diario en *Caponata*

6.10. Información del sistema de *Donald*

Donald es la máquina física destinada a servir como sistema anfitrión de máquinas virtuales. En la página 83. se muestran todas las características del diseño de ésta máquina. En los siguientes apartados se verán los resultados obtenidos por ella.

6.10.1. Información general

En las figuras 6.22 y 6.23, se ven las gráficas obtenida por la herramienta Phpsysinfo. Esta figura se divide en varios apartados que se tratarán a continuación:

- **Vitales**: en éste apartado se puede comprobar la versión del *kernel* que está ejecutando la máquina (*2.6.18.8-xen*), que se trata de una distribución Debian, el tiempo que lleva arrancada desde el último reinicio, y la IP (163.117.148.242), entre otros datos.

- **Información del hardware**: la información que muestra es la relativa a sí misma, dado que es la máquina anfitriona de máquinas virtuales Xen. Se trata de un Intel Core 2 Duo a 2.13 *Ghz*. Es importante resaltar la aparición de los puertos serial ata donde van insertados los 8 discos duros que posee la máquina (en la figura aparecen como puertos SCSI).

- **Utilización de la red**: muestra información relativa al uso de cada interfaz de red. Dado que Donald ejecuta 4 máquinas virtuales, aparecen los interfaces de red virtuales de las 4 máquinas ejecutadas.

- **Utilización de la memoria**: de los 512Mb de memoria RAM asignados a *Donald*, se comprueba como ésta utiliza un 99 %, obligando a utilizar la memoria *swap*, de la que solo utiliza un 1 %. Por lo tanto, es recomendable aumentar la memoria RAM de *Donald*, evitando que se llegue a utilizar la memoria *swap* y disminuya el rendimiento. No obstante, la ampliación de memoria para *Donald* no es algo crítico, dado que tiene el 99 % de la memoria *swap* sin utilizar, y en caso de necesidad puede recurrir a ella sin llegar a ocuparla entera, según las estadísticas de utilización.

- **Sistemas de archivos**: los sistemas de ficheros propios de *Donald* son el raíz y los sistemas virtuales de tipo *tmpfs o udev*. El sistema raíz de *Donald* está ocupado en un 66 %, quedando 10 *Gigas* libres, por lo tanto no se prevee ningún tipo de incidencia respecto al espacio necesitado.

Información Del Sistema: donald.arcos.inf.uc3m.es (163.117.148.242)

Vitales

Nombre Del Sistema	donald.arcos.inf.uc3m.es
Dirección IP	163.117.148.242
Versión Del Núcleo	2.6.18.8-xen (SMP)
Distro Name	Debian 4.0
Uptime	21 horas 49 minutos
Usuarios Actuales	0
Promedio De Uso	0.15 0.21 0.44
	0%

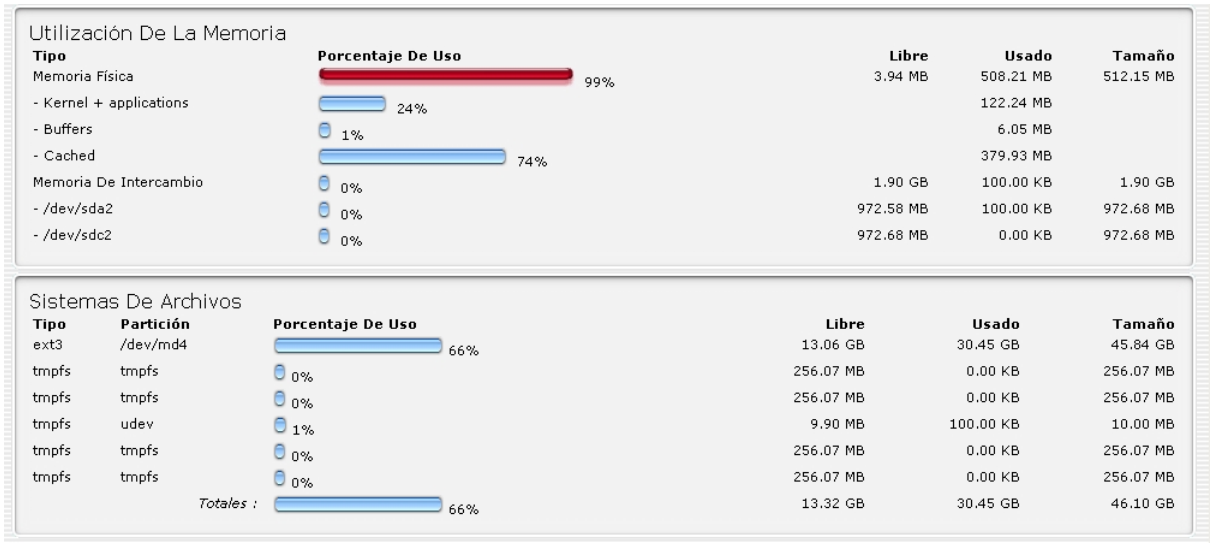
Información Del Hardware

Procesadores	2
Modelo	Intel(R) Core(TM)2 CPU 6420 @ 2.13GHz
Frecuencia	2.13 GHz
Tamaño Del Caché	4096 KB
Bogomips	10642.72
Dispositivos PCI	00:1f.1 IDE interface: Intel Corporation 82801G 00:1f.2 SATA controller: Intel Corporation 82801GR/GH 00:1f.3 SMBus: Intel Corporation 82801G 01:00.1 PIC: Intel Corporation 6700/6702PXH I/OxAPIC Interrupt Controller A 01:00.3 PIC: Intel Corporation 6700PXH I/OxAPIC Interrupt Controller B 09:00.0 RAID bus controller: 3ware Inc Unknown device 1004 0d:00.0 Ethernet controller: Intel Corporation 82573E Gigabit Ethernet Controller 0e:00.0 Ethernet controller: Intel Corporation 82573L Gigabit Ethernet Controller 0f:04.0 VGA compatible controller: ATI Technologies Inc ES1000
Dispositivos IDE	hda: HL-DT-ST DVDRAM GSA-H42N
Dispositivos SCSI	AMCC 9650SE-8LP DISK (Direct-Access) AMCC 9650SE-8LP DISK (Direct-Access) AMCC 9650SE-8LP DISK (Direct-Access) AMCC 9650SE-8LP DISK (Direct-Access) AMCC 9650SE-8LP DISK (Direct-Access) AMCC 9650SE-8LP DISK (Direct-Access) AMCC 9650SE-8LP DISK (Direct-Access) AMCC 9650SE-8LP DISK (Direct-Access)
Dispositivos USB	Linux 2.6.18.8-xen ehci_hcd EHCI Host Controller Linux 2.6.18.8-xen uhci_hcd UHCI Host Controller Linux 2.6.18.8-xen uhci_hcd UHCI Host Controller Linux 2.6.18.8-xen uhci_hcd UHCI Host Controller Linux 2.6.18.8-xen uhci_hcd UHCI Host Controller

Utilización De La Red

Dispositivo	Recibidos	Enviados	Errores/Perdidos
lo	17.82 KB	17.82 KB	0/0
vif0.0	12.24 MB	26.24 MB	0/0
eth0	26.24 MB	12.24 MB	0/0
vif0.1	0.00 KB	0.00 KB	0/0
veth1	0.00 KB	0.00 KB	0/0
vif0.2	0.00 KB	0.00 KB	0/0
veth2	0.00 KB	0.00 KB	0/0
vif0.3	0.00 KB	0.00 KB	0/0
veth3	0.00 KB	0.00 KB	0/0
peth0	1.40 GB	2.67 GB	0/0
eth1	0.00 KB	0.00 KB	0/0
sit0	0.00 KB	0.00 KB	0/0
xenbr0	15.58 MB	0.00 KB	0/0
vif1.0	547.37 MB	2.04 GB	0/470
vif2.0	24.29 MB	40.39 MB	0/164
vif3.0	1.49 GB	274.19 MB	0/3447
vif4.0	3.18 GB	2.00 GB	0/1130

Figura 6.22: Información general sobre *Donald* (1ª parte)



Utilización De La Memoria

Tipo	Porcentaje De Uso	Libre	Usado	Tamaño
Memoria Física	99%	3.94 MB	508.21 MB	512.15 MB
- Kernel + applications	24%		122.24 MB	
- Buffers	1%		6.05 MB	
- Cached	74%		379.93 MB	
Memoria De Intercambio	0%	1.90 GB	100.00 KB	1.90 GB
- /dev/sda2	0%	972.58 MB	100.00 KB	972.68 MB
- /dev/sdc2	0%	972.68 MB	0.00 KB	972.68 MB

Sistemas De Archivos

Tipo	Partición	Porcentaje De Uso	Libre	Usado	Tamaño
ext3	/dev/md4	66%	13.06 GB	30.45 GB	45.84 GB
tmpfs	tmpfs	0%	256.07 MB	0.00 KB	256.07 MB
tmpfs	tmpfs	0%	256.07 MB	0.00 KB	256.07 MB
tmpfs	udev	1%	9.90 MB	100.00 KB	10.00 MB
tmpfs	tmpfs	0%	256.07 MB	0.00 KB	256.07 MB
tmpfs	tmpfs	0%	256.07 MB	0.00 KB	256.07 MB
	Totales :	66%	13.32 GB	30.45 GB	46.10 GB

Figura 6.23: Información general sobre *Donald* (2ª parte)

6.10.2. Uso de disco

En la figura 6.24 se observan las estadísticas de uso diario de los discos de *Donald*. Dado que *Donald* posee 8 discos duros Serial Ata, y varios RAID creados con ellos, los datos estadísticos se pueden agrupar en grupos de dos discos. No obstante, hay que mati-

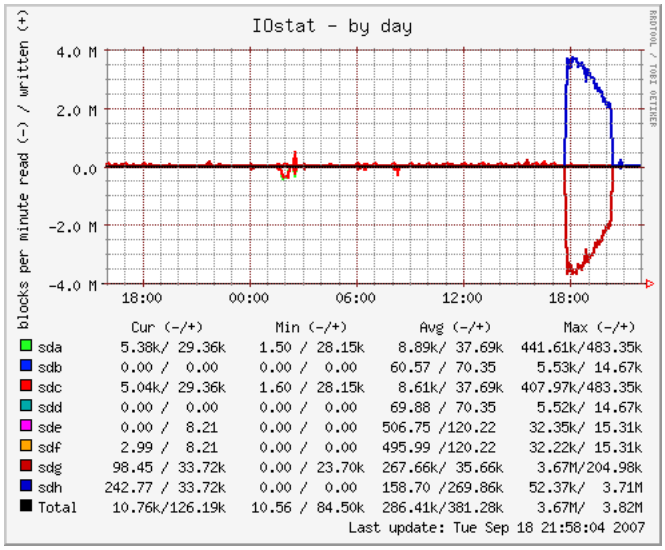


Figura 6.24: Uso diario de disco en *Donald*

zar que a la hora de realizar la captura, uno de los discos no funcionaba, siendo repuesto a las 17:45 aproximadamente. Dado que formaba parte de un RAID 1, en el momento de reposición del disco, se empezó a sincronizar con el otro disco espejo. En la gráfica se puede observar cómo los discos *sdg* y *sdh* transfieren información de uno a otro (*sdg* lee información que se escribe sobre *sdh*, el disco repuesto). La sincronización duró desde las 17:45 hasta las 20:20 horas aproximadamente.

Los RAID de tipo 1 de *Donald*, utilizan los discos de la siguiente forma:

- *sda* con *sdc*.
- *sdh* con *sdg*.
- *sde* con *sdf*.
- *sdb* con *sdd*.

En la gráfica se pueden observar como los valores en los discos duros que componen cada RAID son muy similares. Solo el caso de los discos *sdg* y *sdh* muestra unas diferencias mas notables, dada el fallo de *sdh* y la resincronización posterior.

6.10.3. Tráfico de red

En la figura 6.25 se puede observar el tráfico de red diario del interfaz *eth0* de Donald. Es importante matizar que estas estadísticas solo incluyen el tráfico de *Donald*, no el del resto de máquinas virtuales, cuyos interfaces de red aparecen en *Donald* con otros nombres. El interfaz *eth0* que aparece aqui, es en realidad el interfaz virtual de la máquina anfitriona (el propio *Donald*). El verdadero interfaz *eth0* es la unión de todos los interfaces virtuales, tanto el de la máquina anfitriona, como el resto de máquinas virtuales.

Lo más relevante de lo que aparece en la gráfica es el pico que se produce a las 03:00 horas. Se genera una salida de datos con una transferencia máxima de 3,53 *Mb* por segundo. Esto se produce porque a esa hora está programado el *backup* del sistema raíz de *Donald* a *Lucas*. La transferencia de datos se produce a nivel interno en la máquina, dado que *Lucas* está ejecutándose como máquina virtual en el propio *Donald*, pero el sistema operativo, como se ha comentado antes, utiliza para *Donald* un interfaz virtual, que él ve como *eth0*.

Con lo observado en la gráfica se puede concluir que el tráfico en la máquina *Donald* no es denso, por lo que no se producirá una pérdida de paquetes.

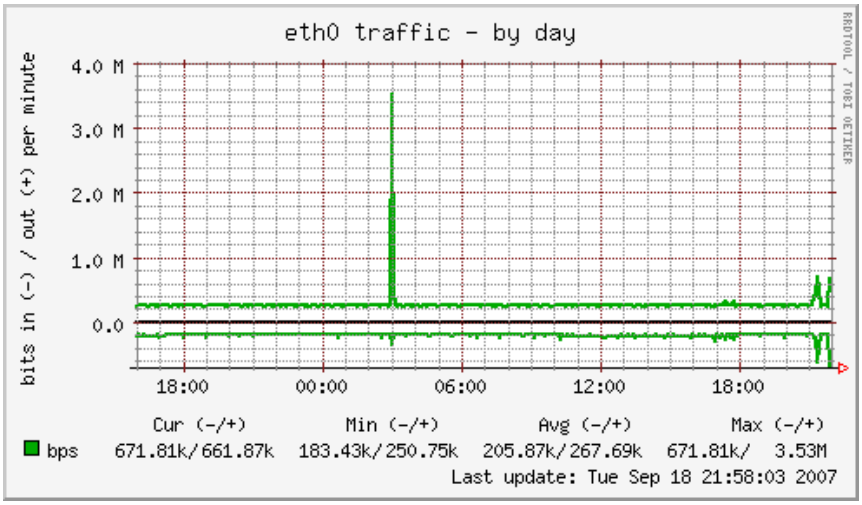


Figura 6.25: Trafico diario por el interfaz *eth0* en *Donald*

6.10.4. Uso de CPU

En la figura 6.26 se observa la utilización diaria de la CPU de *Donald*. El primer hecho relevante que se puede observar es cómo la gráfica tiene como escala el 200 % en lugar del 100 % que se utiliza en las demás máquinas. La razón de esta elección para la escala

es que *Donald* es una máquina con doble nucleo, y por tanto se refleja esto mismo en la gráfica.

Se puede observar como la mayor parte del tiempo la CPU (CPU como procesador de doble núcleo) esta desocupada. El resto de tiempo de la CPU se emplea en tiempo de espera para la entrada y salida. Esto se produce porque *Donald* posee los 8 discos duros que contienen los datos que utiliza todo el sistema, además de las máquinas virtuales, produciendo muchos accesos a disco.

A las 02:30 aproximadamente, se produce un pico de tiempo de espera de entrada y salida. A esa hora es cuando se realiza el *backup* de la máquina *Piojito*, posiblemente el *backup* más costoso en cuanto a accesos a disco, dado que es la máquina virtual cuyo sistema posee más ficheros.

De la gráfica se concluye que los accesos a disco por parte de las máquinas virtuales, y los RAID que existen a un nivel inferior, no disminuyen el rendimiento de *Donald* significativamente, es más, *Donald* mantiene un rendimiento óptimo.

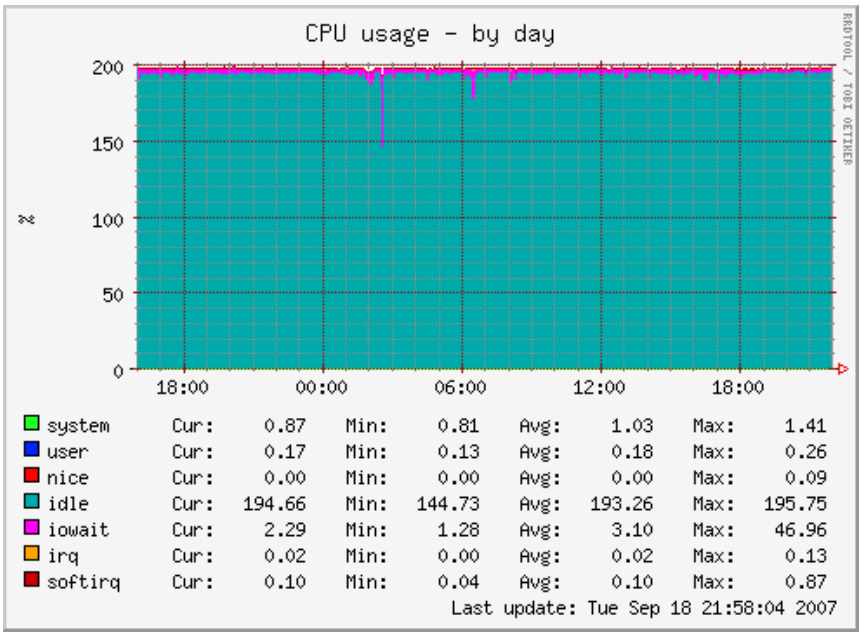


Figura 6.26: Uso de CPU diario en *Donald*

6.11. Información del sistema de *Lucas*

Lucas es la máquina virtual destinada a exportar los datos de las cuentas de usuario a todo el sistema y además realizar las copias de seguridad. En la página 98. se muestran todas las características del diseño de ésta máquina. En los siguientes apartados se verán los resultados obtenidos por ella.

6.11.1. Información general

En la figura 6.27, se ve la gráfica obtenida por la herramienta Phpsysinfo. Esta figura se divide en varios apartados que se tratarán a continuación:

- **Vitales**: en éste apartado se puede comprobar la versión del *kernel* que está ejecutando la máquina (*2.6.18.8-xen*), que se trata de una distribución Debian, el tiempo que lleva arrancada desde el último reinicio, y la IP (163.117.148.244), entre otros datos.

- **Información del hardware**: la información que muestra es la relativa a la máquina física donde se ejecuta *Lucas*, siendo en este caso *Donald* (Intel Core 2 Duo).

- **Utilización de la red**: muestra información relativa al uso de cada interfaz de red (*lo* y *eth0*).

- **Utilización de la memoria**: de los 1024Mb de memoria RAM asignados a *Lucas*, se comprueba como ésta utiliza un 99 %, obligando a utilizar la memoria *swap*, de la que solo utiliza menos de un 1 %. Por lo tanto, es recomendable aumentar la memoria RAM de *Lucas*, evitando que llegue a utilizar la memoria *swap* y disminuya el rendimiento. No obstante, la ampliación de memoria para *Lucas* no es algo crítico, dado que tiene el 99 % de la memoria *swap* sin utilizar, y en caso de necesidad puede recurrir a ella sin llegar a ocuparla entera, según las estadísticas de utilización.

- **Sistemas de archivos**: *Lucas* tiene acceso a los volúmenes lógicos creados en *Donald*, dado que es la máquina virtual que los utiliza. Para *Lucas*, estos volúmenes lógicos se comportan como discos duros, que se corresponden con los dispositivos *sda3*, *sda4* y *sda5*. El dispositivo *sda1* es el fichero imagen de *Lucas*, que contiene su sistema raiz y está ocupado en un 14 %.

  Los dispositivos *sda3*, *sda5* y *sda5* se corresponden con los volúmenes lógicos: *sistema*, *usuarios* y *backup_usuarios*, respectivamente. El volumen lógico *sistema* (dispositivo *sda3*), es el que más ocupado esta, con un 59 % de su totalidad. Aunque supera la mitad de espacio asignado, en ocupación, no se prevee un incremento significativo, dado que almacena copias de seguridad del sistema y va eliminando las copias antiguas.

El volumen lógico *usuarios* (dispositivo *sda4*), está ocupado en un 29 % de su totalidad, siendo éste un valor aceptable. Por último, el volúmen lógico *backup_usuarios* (dispositivo sda5) esta ocupado en un 15 %, siendo también un valor aceptable.

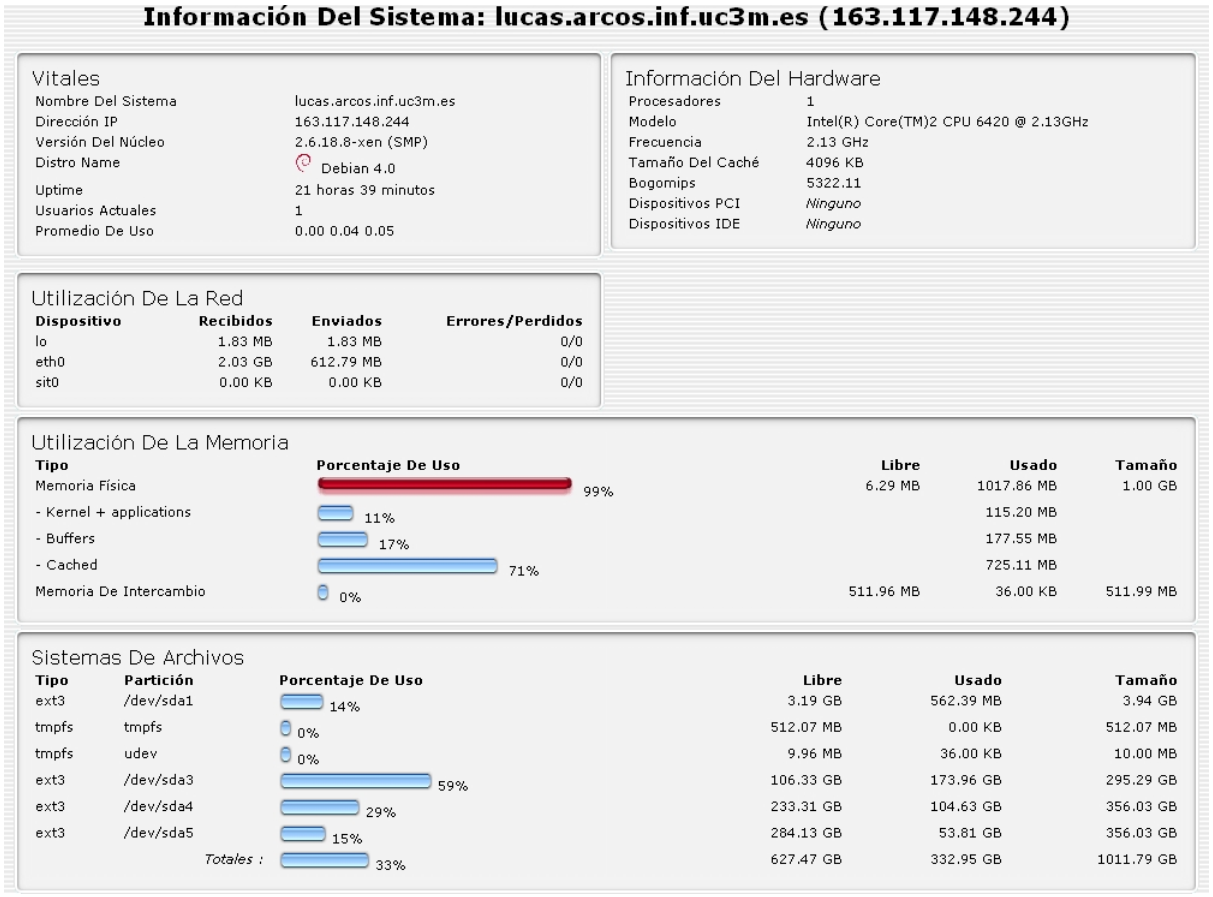


Figura 6.27: Información general sobre *Lucas*

6.11.2. Servidor NFS

En la figura 6.28 se puede observar la utilización del servidor NFS de *Lucas*. Lo más importante que se puede observar es el siguiente hecho: si se comprueba ésta gráfica con la de la figura 6.33 en la página 203, se puede observar como son casi idénticas. Ésta última figura se corresponde con la utilización del cliente NFS en *Piojito*, y de la comparación de ambas figuras se extraen varias conclusiones que se analizan a continuación:

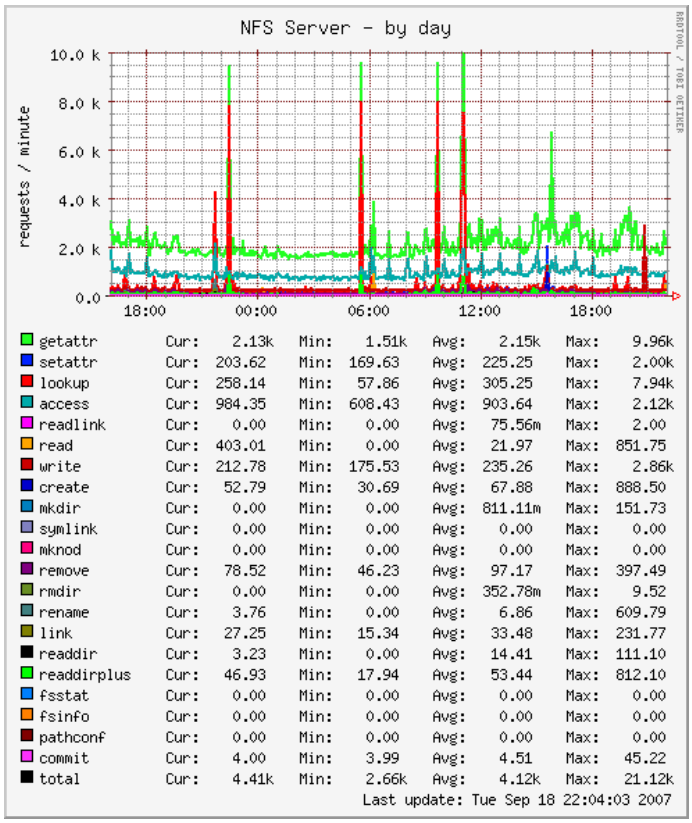


Figura 6.28: Uso diario del servidor NFS en *Lucas*

Dado que Piojito actúa como cliente NFS importando sistemas de ficheros de *Lucas*, y utiliza dichos sistemas para los servicios de web y correo, cada vez que se haga uso de estos servicios se generarán estadísticas en ambas gráficas. En el caso de *Lucas* se añadirán a sus estadísticas, las generadas por el uso que haga tanto *Piojito* como el resto de las máquinas virtuales. No obstante, como la máquina *Piojito* es la que más utiliza el servidor NFS, ambas gráficas se parecerán notablemente.

Los picos que aparecen en la figura corresponderán a accesos provocados por el servidor de correo, dado que tiene que tratar con muchos ficheros en cada buzón de correo (al utilizar Maildirs, hay un fichero por cada correo).

También se puede observar una diferencia respecto a la gráfica de la máquina *Piojito*. A las 21:00 horas se produce un pico de accesos de escritura provenientes de *Caponata*, tal y como se puede observar en la figura 6.20 en la página 188, donde se observa la gráfica de *Caponata* como cliente NFS del servidor *Lucas*. Este pico demuestra que la gráfica de *Lucas*, como servidor de NFS, reflejará todas las estadísticas individuales que tengan las máquinas que importen sistemas de ficheros de *Lucas*.

6.11.3. Tráfico de red

En la figura 6.29 se puede observar el tráfico del interfaz *eth0* en *Lucas*. Se observan varios picos que se corresponden con las copias de seguridad y con ficheros copiados a *Caponata*. A continuación se explican estas conclusiones:

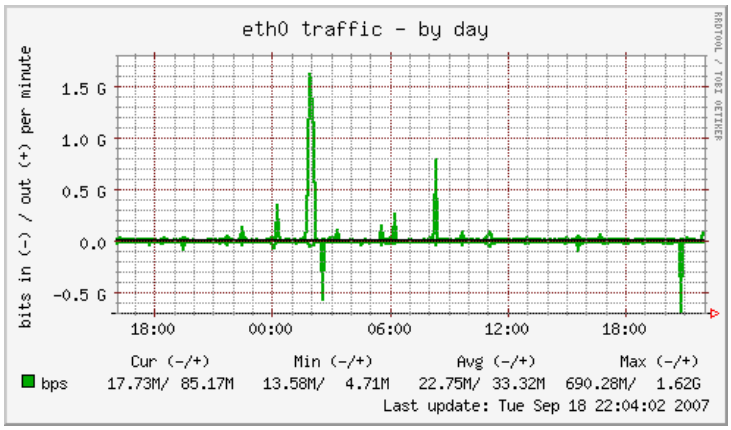


Figura 6.29: Trafico diario por el interfaz *eth0* en *Lucas*

El pico que empieza a las 01:30 se corresponde con una copia de seguridad que se hizo a una máquina distinta de *Boyerito*, para realizar ciertas pruebas.

El pico que aparece a las 02:30 se corresponde con la copia de seguridad del sistema raíz de *Piojito*. Como es la máquina virtual que más ficheros contiene y más cambios se producen dentro de ella, se realiza la transferencia de muchos datos al espacio de *backup* de *Lucas*.

La copia de seguridad de los espacios de almacenamiento (*usuarios*, *backup_usuarios* y *sistema*) a *Boyerito* se produce a las 06:00, 07:00 y 08:00 horas, siendo la copia de las 08:00 horas la que más ficheros tiene que transferir. Ésta última transferencia es del espacio denominado *sistema*, donde se han realizado todas las copias de seguridad de las máquinas y los servicios, por tanto, cada dia tiene un número elevado de datos nuevos que transferir. Los espacios *usuarios* y *backup_usuarios*, no tienen tantos cambios, generando picos menores o casi imperceptibles.

Por último, aparece un pico de entrada de datos por el interfaz a las 21:00 horas. Esta entrada de datos se produce porque algún usuario transfirió datos a *Caponata*, y dado que el espacio de ficheros donde se transfirieron dichos datos, estaba importado por NFS desde *Lucas*, realmente se transferían a *Lucas*. Esta transferencia se puede observar tanto en la gráfica del servidor NFS de *Lucas* (ver página 197.), como en la gráfica de cliente NFS de *Caponata* (ver página 188.).

6.11.4. Uso de CPU

En la figura 6.30 se puede observar la utilización diaria de la CPU de *Lucas*. Se observa que la mayor parte del tiempo, la CPU esta desocupada, siendo en porcentaje un 97,38 % respecto a las últimas 24 horas a partir de la fecha de captura de la gráfica.

También se observa como el tiempo restante de CPU lo ocupan mayoritariamente los tiempos de espera de dispositivos de entrada/salida, y los tiempos utilizados para procesos con la prioridad modificada con el comando *nice*.

Se puede observar como hay varios picos que se producen desde las 0:00 horas hasta las 08:20 horas aproximadamente. Estos picos se producen cuando se realizan las copias de seguridad con *rsync*:

- Se produce en primer lugar un pico de tipo *iowait*. Se realiza un acceso a disco para introducir en cache una copia de la estructura de ficheros.
- A continuación comienza la copia de ficheros producido por el comando *rsync*. La prioridad de este comando es modificada con *nice*, por eso aparecen las líneas de color rojo en la gráfica.

Los picos de tipo *iowait* siempre serán elevados, mientras que los picos de tipo *nice* variarán según los cambios producidos diariamente en el sistema. Es decir, cada vez que se realice un *rsync* desde *Lucas*, se tendrá que copiar a memoria la estructura de ficheros de la parte que esta localizada en *Lucas*, y una vez que empiece la sincronización, los datos a transferir dependerán de los cambios diarios que haya en el sistema. Si no se han producido apenas cambios, la sincronización de datos será mínima (y por tanto los picos de tipo *nice*).

Con todo esto se concluye que el mayor gasto de CPU en *Lucas* se produce por los accesos a disco.

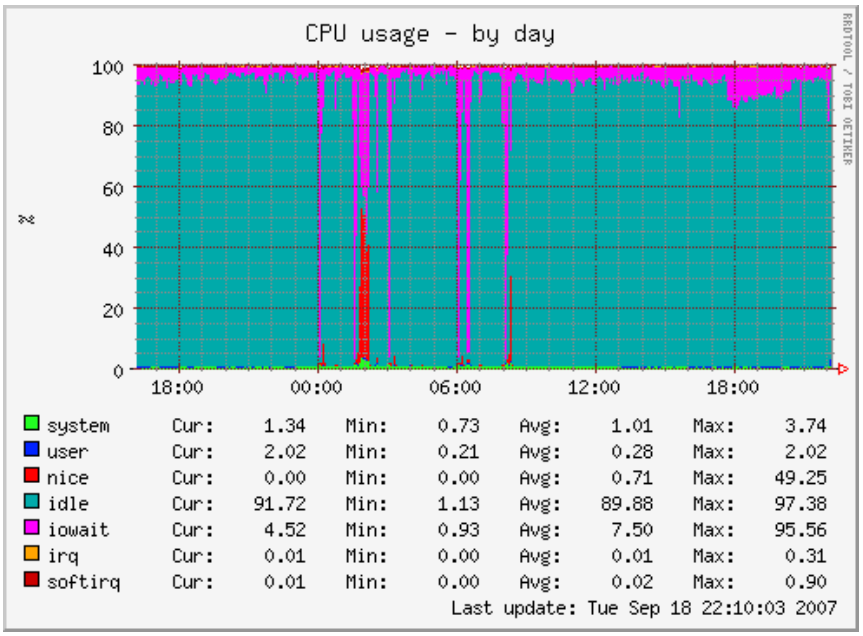


Figura 6.30: Uso de CPU diario en *Lucas*

6.12. Información del sistema de Piojito

Piojito es la máquina virtual destinada a ofrecer los servicios externos del sistema, como la web y el correo. En la página 107. se muestran todas las características del diseño de ésta máquina. En los siguientes apartados se verán los resultados obtenidos por ella.

6.12.1. Información general

En la figura 6.31, se ve la gráfica obtenida por la herramienta Phpsysinfo. Esta figura se divide en varios apartados que se tratarán a continuación:

- **Vitales**: en éste apartado se puede comprobar la versión del *kernel* que está ejecutando la máquina (*2.6.18.8-xen*), que se trata de una distribución Debian, el tiempo que lleva arrancada desde el último reinicio, y la IP (163.117.148.240), entre otros datos.

- **Información del hardware**: la información que muestra es la relativa a la máquina física donde se ejecuta *Piojito*, siendo en este caso *Donald* (Intel Core 2 Duo).

- **Utilización de la red**: muestra información relativa al uso de cada interfaz de red (*lo* y *eth0*).

- **Utilización de la memoria**: de los 1490Mb de memoria RAM asignados a *Piojito*, se comprueba como ésta utiliza un 91 %, llegando a utilizar la memoria *swap*, de la que utiliza menos de un 1 %. Por lo tanto, es recomendable aumentar la memoria RAM de *Piojito*, para que no llegue a utilizar la memoria *swap* y disminuya el rendimiento. No obstante, la ampliación de memoria para *Piojito* no es algo crítico, dado que tiene más del 99 % de la memoria *swap* sin utilizar, y en caso de necesidad puede recurrir a ella sin llegar a ocuparla entera, según las estadísticas de utilización.

- **Sistemas de archivos**: los sistemas de ficheros propios de *Piojito* son el raíz y el sistema virtual */dev/shm*. El sistema raíz de *Piojito* está ocupado en un 51 %, por lo tanto no se prevee ningún tipo de incidencia respecto al espacio necesitado.

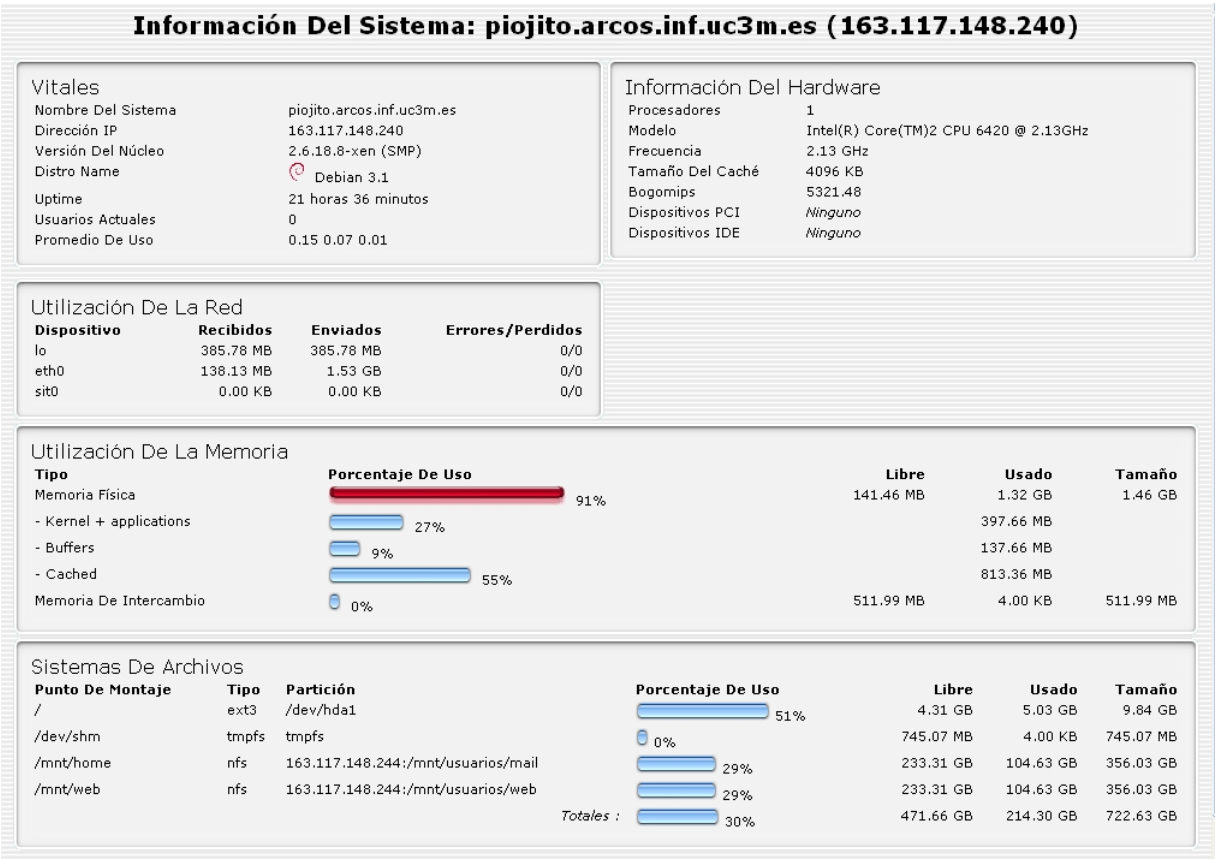


Figura 6.31: Información general sobre *Piojito*

6.12.2. Tráfico de red

En la figura 6.32 se observa el comportamiento del interfaz *eth0* diariamente. Se pueden ver transferencias máximas entorno a los 500*Mbps*, correspondientes a los picos que aparecen en la gráfica.

Varios de los picos se corresponden a acciones diarias de manteniemiento, apareciendo en las gráficas que se obtengan cada día. El primero de estos picos es el de las 02:30 horas, correspondiente al *backup* del sistema raíz de *Piojito*, y realizado desde la máquina virtual *Lucas*. Se observa en la gráfica como el pico está en el lado de *bits* de salida del interfaz *eth0*, dado que es *Piojito* quien envía sus datos a *Lucas*.

Los otros picos que aparecen en la figura, y que corresponden a tareas de mantenimiento, son los que aparecen desde las 06:00 a las 18:00 horas. Estos picos se producen por la monitorización realizada con Snort, que ejecuta un *script* en el dicho tramo horario, cada hora. Este *script* recibe información de la cuenta *'web'* a través de la máquina *Caponata*, después analiza dicha información mediante Snort y genera varias páginas *html* que devuelve a la cuenta *'web'*, de nuevo a través de *Caponata*. Lo que refleja la gráfica es la recepción de varios *megabits* de datos a través de *Caponata*, pero el envio de unos pocos *Kbs*, que apenas se perciben. En la figura 6.19 en la página 187, se puede observar como la máquina *Caponata* también muestra la transferencia de datos del Snort, pero en este caso, saliendo del interfaz *eth0*.

El resto de picos que aparecen en la gráfica se deberán a los servicios externos que integra *Piojito* (como web y correo).

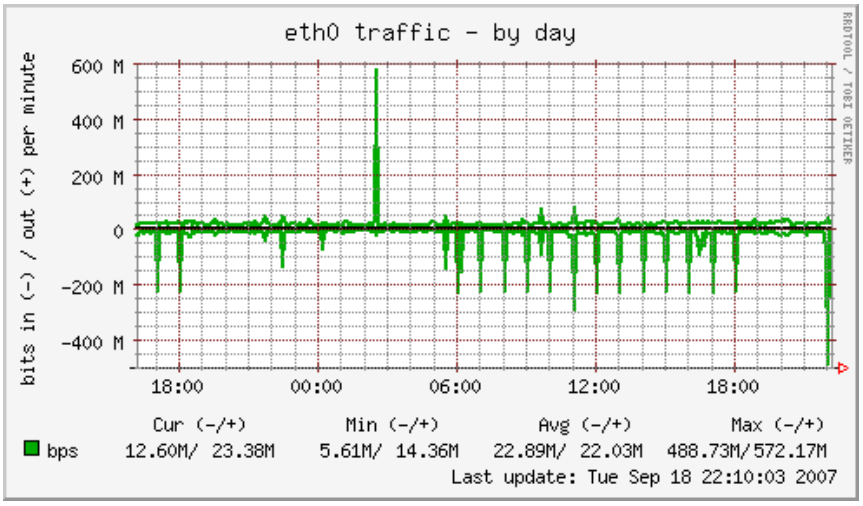


Figura 6.32: Trafico diario por el interfaz *eth0* en *Piojito*

6.12.3. Cliente NFS

En la figura 6.33 se puede observar la utilización diaria del cliente NFS de *Piojito*. Los sistemas de ficheros importados pertenecen a *Lucas*, y corresponden a los directorios */mnt/mail* y */mnt/web* utilizados por el servidor de correo y el servidor web.

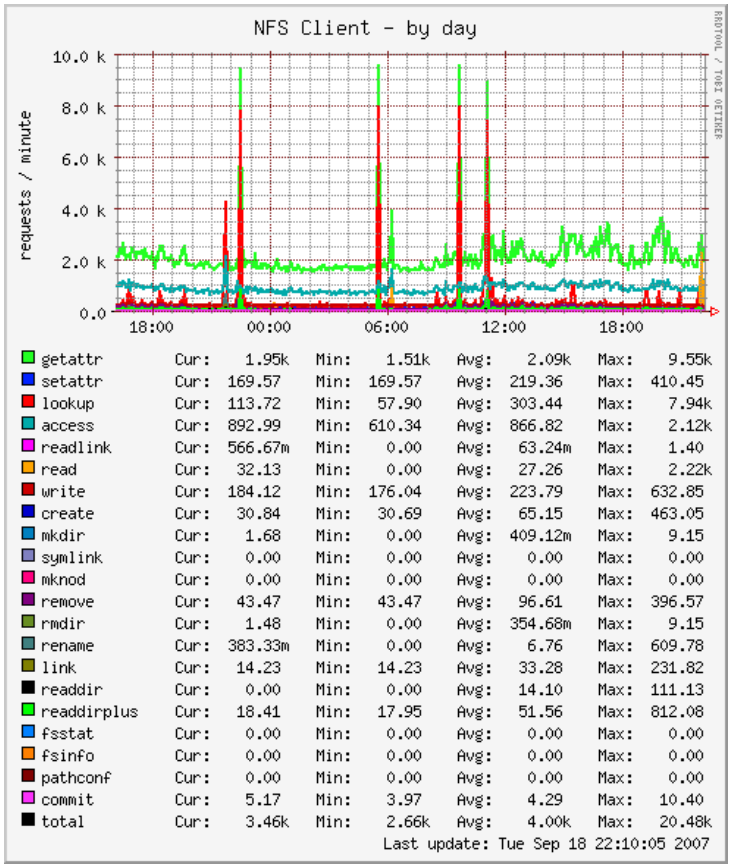


Figura 6.33: Cliente NFS diario en *Piojito*

Lo que se puede observar en la gráfica es la utilización de dichos sistemas de ficheros por parte de ámbos servicios (web y correo). Los picos que aparecen se producen por accesos a directorios con muchos ficheros, dado que son de tipo *lookup* y *getattr* principalmente. Estos accesos son provocados por el acceso a buzones de correo con múltiples ficheros (el servidor de correo utiliza el formato Maildir, un fichero por cada correo), como se da en las cuentas de usuario de personas que reciben y almacenan diariamente una cantidad notable de correos.

Uno de los hechos más importantes que hay que destacar, es que *Piojito* es la máquina virtual que más utiliza el servidor NFS de *Lucas*. Esto se refleja al contemplar la

gráfica de *Lucas* como servidor NFS (ver figura 6.28 en página 197.), y la gráfica de *Piojito* como cliente NFS (ver figura 6.33). Ambas gráficas son muy parecidas, siendo la gráfica de *Lucas* más completa, dado que contiene todas las peticiones de *Piojito* y del resto de máquinas virtuales.

De la gráfica se concluye que el sistema llega a atender hasta 9.500 peticiones por segundo en los picos más altos, pero ésto se produce pocas veces al día. El resto del tiempo, las peticiones NFS se mantienen estables, entorno a las 2000 peticiones por segundo. Por tanto, el sistema ofrece un rendimiento óptimo respecto a la utilización del servicio NFS.

6.12.4. Uso de CPU

En la figura 6.34 se puede observar la utilización diaria de la CPU de *Piojito*. Lo más destacable es que la mayor parte del tiempo la CPU esta desocupada, alrededor de un 96,09 % en las últimas 24 horas respecto al dia de la captura.

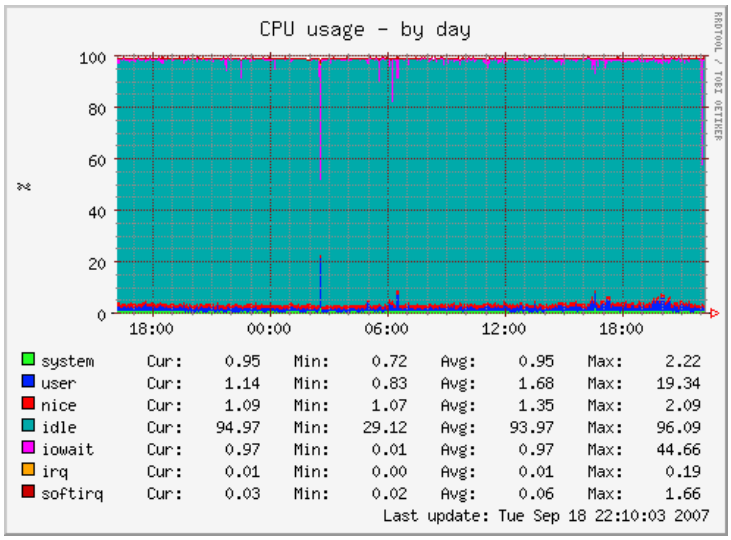


Figura 6.34: Uso de CPU diario en *Piojito*

El resto del tiempo la ocupación de la CPU se divide principalmente en procesos ejecutados a nivel de usuario y peticiones de entrada/salida. Se puede también observar como a las 02:30, cuando se produce la copia de seguridad del sistema raíz de *Piojito*, hay un pico de tipo *iowait* y de tipo *user*. La copia se realiza utilizando *rsync*, que abre una conexión SSH con Piojito, por lo tanto, en la gráfica se observan los accesos a disco (*iowait*) y la transferencia utilizando SSH (*user*).

6.13. Modificaciones realizadas

Con motivo del análisis del resultado de las gráficas de monitorización del sistema, se han realizado ciertas modificaciones sobre él. Las modificaciones más importantes han sido ajustar la memoria de cada máquina virtual, dados los 4 *Gigas* de que dispone *Donald*.

Por ejemplo, Piojito inicialmente tenía sólo 1 *Giga* de Ram, y *Donald* (como sistema anfitrión *Dom0*) tenía 1 *Giga* también. Lo que se hizo fue reducir la memoria de *Donald*, ya que no consumía tanta, a 512 *Mb*, y añadir 512 *Mb* a *Piojito*.

Otro de los cambios ha sido en uno de los *plugins* de Munin (*plugin auth*), comentado en el siguiente apartado, así como la instalación de la herramienta DenyHosts.

6.13.1. Plugin *auth* de Munin

Inicialmente la gráfica que se obtenía del *plugin auth* de Munin instalado en *Caponata* era la que aparece en la figura 6.35.

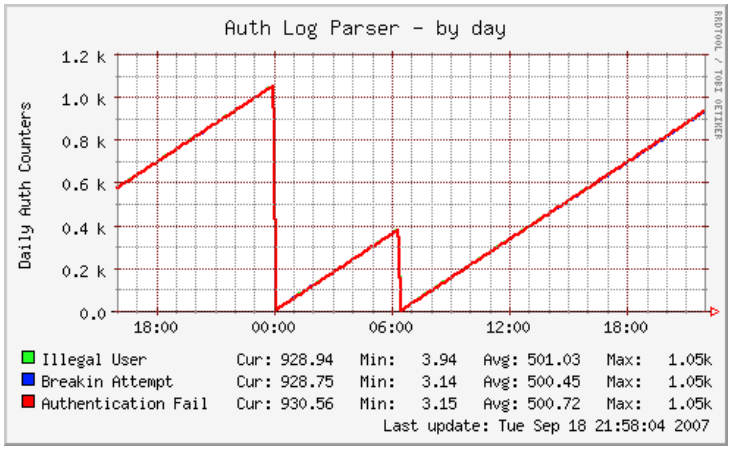


Figura 6.35: Intrusiones diarias en *Caponata*

Esta gráfica muestra un comportamiento dudoso respecto a los intentos de ataque hacia *Caponata*. En primer lugar, la periodicidad con que aparecen los supuestos ataques es motivo de duda, dado que éstos suelen darse de forma más aleatoria. En segundo lugar, los ataques suelen ser de un solo tipo, y según muestran los datos de la gráfica, se dan los tres parámetros casi al mismo tiempo: *Illegal User*, *Breakin Attempt* y *Authentication Fail*. Y por último, la curva creciente que se produce desde el supuesto inicio al supuesto final: en un ataque real, el numero de intentos por segundo suele ser constante, nunca creciente.

Todas estas observaciones ponían en duda la autenticidad de los resultados del *plugin*, motivando a la revisión del mismo. El *script* de Munin *auth*, tiene el siguiente código:

```
#!/bin/sh
#
#
# Script to show auth stuff
#
# Parameters understood:
#
#       config   (required)
#       autoconf (optional - used by munin-config)
#
#
# Magic markers (optional - used by munin-config and installation
# scripts):
#
#%# family=auto
#%# capabilities=autoconf

MAXLABEL=20

if [ "$1" = "autoconf" ]; then
        echo yes
        exit 0
fi

if [ "$1" = "config" ]; then

        echo 'graph_title Auth Log Parser'
        echo 'graph_args --base 1000 -l 0'
        echo 'graph_vlabel Daily Auth Counters'
        echo 'illegal_user.label Illegal User'
        echo 'possible_breakin.label Breakin Attempt'
        echo 'authentication_failure.label Authentication Fail'
        exit 0
fi

echo -en "illegal_user.value "
echo $(grep "Illegal user\|no such user" /var/log/auth.log \
                        | grep "`date '+%b %d'`" | wc -l)
echo -n
echo -en "possible_breakin.value "
echo $(grep -i "breakin attempt" /var/log/auth.log | grep "`date '+%b %d'`" | wc -l)
```

```
echo -en "authentication_failure.value "
echo $(grep "authentication failure" /var/log/auth.log | grep "‘date ’+%b %d’‘" \
                                                          | wc -l)
```

Se puede comprobar como utiliza el comando *grep* para obtener los intentos de ataque del fichero */var/log/auth.log*. Para comprobar la efectividad del *plugin* se ejecutó a mano la primera de las líneas que obtienen la información (sin el comando *wc*, que cuenta el número de líneas), y se observó el resultado:

```
# grep -i "breakin attempt" /var/log/auth.log | grep "‘date ’+%b %d’‘"
....
Sep 23 14:46:35 ssh snoopy[8913]: [unknown, uid:0 sid:8911]:
                            grep Illegal user\|no such user\|Inv
Sep 23 14:47:35 ssh snoopy[9242]: [unknown, uid:0 sid:9240]:
                            grep Illegal user\|no such user\|Inv
Sep 23 14:48:35 ssh snoopy[9571]: [unknown, uid:0 sid:9569]:
                            grep Illegal user\|no such user\|Inv
Sep 23 14:49:35 ssh snoopy[9900]: [unknown, uid:0 sid:9898]:
                            grep Illegal user\|no such user\|Inv
....
```

El resultado confirmó los falsos datos que se obtenían de este plugin. Lo que aparece es un fragmento de todas las líneas que aparecieron al realizar la búsqueda con los parámetros del *plugin*. Se puede observar como el *plugin* no ha tenido en cuenta la instalación de la librería Snoopy, que registra en el fichero */var/log/auth.log* los comandos ejecutados en la máquina. Dado que el *plugin* realiza un *grep* sobre éste fichero buscando las cadenas *Illegal user* o *no such user*, lo que encuentra en éste fichero es la propia búsqueda, contabilizándola como si de un ataque se tratase. Esto explica varias cosas:

- La curva ascendente que se produce en la gráfica se debe a que cada vez que se realiza una búsqueda en el fichero */var/log/auth.log*, hay más resultados positivos, ya que contabiliza las mismas búsquedas, que van creciendo en número.
- La curva termina bruscamente a las 00:00, y como la búsqueda de la cadena en el fichero *auth.log* se realiza para el día en curso, cuando llegan las 00:00 no encuentra ninguna cadena. También termina bruscamente a las 06:25, cuando se ejecuta el comando *logrotate*, que rota el fichero */var/log/auth.log* en */var/log/auth.log.0.gz*, dejando el fichero *auth.log* vacío.
- Hay el mismo número de intentos para *Illegal User*, *Breakin Attempt* y *Authentication Fail* porque encuentra estas cadenas en el fichero *auth.log* de las búsquedas anteriores.

El siguiente paso a realizar es modificar el *plugin* para que realice búsquedas certeras. En primer lugar hay que eliminar del patrón de búsqueda los resultados que contengan la

palabra *snoopy* (con grep -v "*snoopy*"), también hay que añadir las cadenas que utiliza el fichero *auth.log* en este caso (*Invalid user*, *invalid user*, *BREAK-IN ATTEMPT* y *[Aa]uthentication failure*) y por último realizar la búsqueda en todos los ficheros *auth.log*, incluyendo los rotados. Para este último punto es necesario que los ficheros rotados de *auth.log* no estén comprimidos. Para ello se descomprimen manualmente y se indica en el fichero de configuración del comando *logrotate* que a partir de ese momento realice la rotación sin comprimirlos (solo para el *auth.log*).

El resultado final fue cambiar las últimas líneas del *plugin* por las siguientes:

```
echo -en "illegal_user.value "
echo $(grep "Illegal user\|no such user\|Invalid user\|invalid user" \
        /var/log/auth.log* | grep "`date '+%b %d'`" | grep -v "snoopy" | wc -l)
echo -n
echo -en "possible_breakin.value "
echo $(grep "BREAK-IN ATTEMPT" /var/log/auth.log* |  grep -v "snoopy"| \
                            grep "`date '+%b %d'`" | wc -l)
echo -en "authentication_failure.value "
echo $(grep "uthentication failure" /var/log/auth.log* |  grep -v "snoopy" \
                           | grep "`date '+%b %d'`" | wc -l)
```

Dando valores más certeros, como se puede comprobar en la figura 6.36, donde aparecen 3 intentos ilegales de conexión y 4 autenticaciones fallidas.

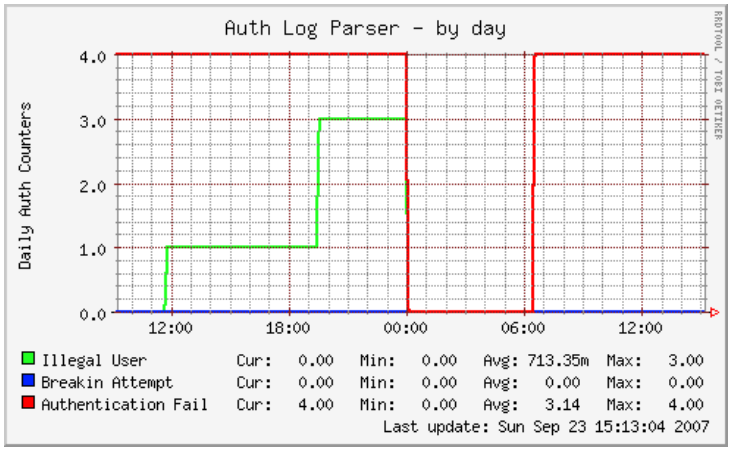


Figura 6.36: Intrusiones diarias en *Caponata* (*plugin* revisado)

6.13.2. Instalación de DenyHosts

Dados los intentos ilegales de acceso a la máquina *Caponata* que se han registrado, se pensó en buscar una solución que permitiese banear las direcciones IP de las máquinas

que realizaban dichos intentos. La solución fue el paquete *denyhosts*, un *script* que impide que cualquier ataque por medio de SSH tenga éxito.

Lo que realiza el *script* es un análisis del *log /var/lib/auth.log* donde se registran los intentos de acceso por SSH. La herramienta añade al fichero */etc/hosts.deny* aquellas direcciones IP que realizan varios intentos fallidos.

Aunque la herramienta DenyHosts, es necesaria fundamentalmente en *Caponata*, se ha instalado en todas las máquinas del sistema. Para su instalación, se ejecutó el siguiente comando:

```
# apt-get install denyhosts
```

Y se modifico el fichero */etc/denyhosts.conf* (ver página 305.) para que bloquease no solo el servicio SSH para las direcciones IP que intentasen un ataque, sino todos los servicios.

Capítulo 7

Conclusiones y trabajos futuros

7.1. Presupuesto

En esta sección se mostrará el presupuesto calculado para el proyecto, incluyendo todos los gastos necesarios para su total elaboración y puesta en funcionamiento.

El tiempo que ha requerido la realización del proyecto ha sido de 15 meses, desde Julio del 2006 a Septiembre del 2007, ambos incluidos. En las siguientes tablas se detalla el tiempo empleado en cada actividad que forma el proyecto.

El trabajo ha sido realizado a razón de 4 horas diarias durante los 15 meses especificados. Suponiendo 20 días hábiles de media por cada mes, supone 1320 horas de trabajo (15 meses * 20 días/mes * 4 horas/día = 1200 horas).

7.1.1. Desglose por actividades

La tabla 7.1 muestra las distintas actividades que se llevaron a cabo para la realización del proyecto, incluyendo el número de horas necesarias de cada una de las mismas.

Tarea	**Horas**
Análisis	120h
Diseño	120h
Implantación	600h
Pruebas	240h
Documentación	120h

Cuadro 7.1: Actividades y duración de las mismas a lo largo del proyecto

7.1.2. Salarios por categoría

En la elaboración del proyecto se requiere personal informático cualificado, el cual deberá adoptar determinados roles distintos que se adapten a cada una de las actividades que forman el proyecto. La tabla 7.2 muestra el coste de cada rol utilizado en la elaboración del proyecto.

Los parámetros especiales utilizados en la tabla 7.2 son:

- **Coste/Hora**: indica el sueldo bruto de una hora de trabajo.
- **Sueldo Bruto Año**: indica el sueldo bruto anual con 14 pagas a lo largo del año.

Para los cálculos se ha supuesto lo siguiente:

- La jornada laboral es de 4 horas diarias.
- Son 20 los días laborables del mes.
- Contamos con 1200 horas que suponen 15 meses * 20 dias/mes * 4 horas/dia .

Cargo	Horas	Coste/Hora	Sueldo Bruto año
Analista	120h	25€	28.000€/año
Responsable de diseño	120h	25€	28.000€/año
Responsable de implantación	600h	20€	22.400€/año
Ingeniero de pruebas	240h	25€	28.000 €/año
Responsable de documentación	120h	20€	22.400€/año

Cuadro 7.2: Tabla de salarios

7.1.3. Gastos de personal imputables al proyecto

Este proyecto ha sido realizado por una sola persona, que adoptó los distintos roles que aparecen en la tabla 7.2 para desarrollar cada una de las actividades que forman el proyecto. La tabla 7.3 muestra el coste total de cada rol. El importe total es de 26.400€.

7.1.4. Recursos materiales empleados

La tabla 7.4 muestra el coste de cada recurso material necesario para la implantación del proyecto. Los recursos principales son las máquinas empleadas y el software empleado. Es importante destacar el coste cero del software empleado, dado que se ha utilizado

Cargo	Coste/Hora	Total
Analista	25€	3.000€
Responsable de diseño	25€	3.000€
Responsable de implantación	20€	12.000€
Ingeniero de pruebas	25€	6.000€
Responsable de documentación	20€	2.400€

Cuadro 7.3: Gastos de personal imputables al proyecto

software libre en el presente proyecto para reducir lo máximo posible el coste del mismo.

Respecto a las especificaciones técnicas de cada máquina, estas aparecen en la sección 3.4.1 en la página 72. Los discos duros de cada una de ellas se listarán aparte.

Los costes incluyen el I.V.A, y el total de todos los recursos materiales asciende a 6.175€.

Recurso	Cantidad	Precio	Coste total
Máquina Donald	1	3.000€	3.000€
Máquina Daisy	1	2.000€	2.000€
Máquina Boyerito	1	500€	500€
Software utilizado	-	**0€**	**0€**
Switch Ethernet Gigabit 3Com	1	40€	40€
Cable Ethernet CAT-5	3	5€	15€
Disco duro Serial ATA 250G	6	40€	240€
Disco duro Serial ATA 400G	2	50€	100€
Disco duro ATA 100 250G	4	40€	160€
Disco duro ATA 100 120G	4	30€	120€

Cuadro 7.4: Recursos materiales empleados

7.1.5. Gastos indirectos

El coste de los gastos indirectos se realiza sobre la base de un 10 % del total de los costes del proyecto ascendiendo a (26.400 + 6.175)*10/100 = 3.257,5€.

7.1.6. Resumen del presupuesto

La tabla 7.5 muestra un resumen de todos los costes involucrados en el proyecto, así como la suma total de los mismos.

El concepto margen de riesgo se realiza con un 20 % de la suma de todos los conceptos anteriores: (26.400 + 6.175 + 3.257,5)*20/100 = 7.166,5€. El concepto de beneficio se calcula con un 15 % de todos los conceptos anteriores: (26.400 + 6.175 + 3.257,5 + 7.166,5€)*15/100 = 6.449,85€.

El presupuesto total del proyecto es de 49.448,85€.

Recurso	**Coste total**
Personal con cargo al proyecto	26.400€
Recursos materiales empleados	6175€
Gastos indirectos	3257,5€
Margen de riesgo (20 %)	7.166,5€
Beneficio (15 %)	6.449,85€
Total	49.448,85€

Cuadro 7.5: Presupuesto para la realización del proyecto

7.2. Conclusiones

En este apartado se verán las conclusiones que he obtenido a la finalización del presente proyecto. Se dividen en tres apartados, en los cuales expongo lo siguiente: mis conclusiones sobre la virtualización aplicada a servidores, las conclusiones que he obtenido del sistema implantado y por último las aportaciones que he obtenido durante todo el proyecto.

7.2.1. Conclusiones sobre la virtualización aplicada a servidores

Existen dos problemas en los sistemas de servidores diseñados para los centros de procesamiento de datos: por un lado la proliferación de máquinas físicas, y por otro lado, de manera derivada, el aprovechamiento poco óptimo de la potencia de cómputo de las mismas. A cada máquina física se le asigna una función, instalándose los servicios necesarios para cumplir su objetivo. Si con estos servicios ejecutándose, la máquina cuenta con un 80 % de su potencia de cómputo libre, se está desperdiciando una cantidad considerable de capacidad de proceso en la máquina.

Las máquinas virtuales tienen entre sus ventajas, la capacidad de paliar el problema de la potencia de cómputo no aprovechada. Si el diseño de un sistema de servidores contempla la utilización de varias máquinas enfocadas a distintos objetivos, y para implantar el diseño se utilizan máquinas virtuales ejecutándose sobre una o varias máquinas físicas, se obtendrán varios beneficios:

- Por un lado una reducción de los costes considerable, al no tener que comprar una máquina física por cada máquina del diseño. Se comprarán solo las máquinas físicas que necesite el diseño, aunque tienen que ser de mayor potencia.
- Por otro lado, un aprovechamiento mayor de la potencia de cómputo de las máquinas físicas.
- Otro de los beneficios es la velocidad con la que se comunican las máquinas, que al estar ejecutándose sobre la misma máquina física no dependerán del ancho de banda de una red ethernet por ejemplo, sino del ancho de banda del bus de la propia máquina física.
- Se obtendrá un sistema ampliamente escalable, es decir, si se requieren más máquinas, estás pueden ser máquinas virtuales. De esta forma, no se incurre en más gastos, ni se necesita más espacio físico para alojar otra máquina física.
- Se pueden realizar más divisiones de las que en un principio se contaría al tener solo máquinas físicas. Al realizar más divisiones contando con las máquinas virtuales, se aísla mejor los servicios obteniendo más seguridad en todo el sistema.

Son claros los beneficios que aportan las máquinas virtuales aplicadas al diseño de un sistema de servidores. El presente proyecto utiliza esta innovación tecnológica obteniendo un sistema potente y escalable de servidores, con un bajo coste. Las monitorizaciones realizadas sobre el sistema confirman la calidad del diseño, dado que no hay cuellos de botella y todos los servicios funcionan perfectamente y con una rapidez óptima.

Mi pronóstico de futuro es que existirá una tendencia general a la utilización de sistemas basados en máquinas virtuales en los sistemas de servidores. Hoy en día son comunes las máquinas con procesadores con doble núcleo, incluyendo aquellas destinadas a ser servidores. La virtualización aprovecha la tecnología de estos procesadores, y estos a su vez incluyen instrucciones internas de virtualización, apoyando esta tendencia de uso de máquinas virtuales.

7.2.2. Conclusiones sobre el sistema

A la finalización del presente proyecto, se ha obtenido un sistema de servidores que cumple con los objetivos propuestos. Este sistema obtiene los parámetros deseados: fiabilidad, rapidez, disponibilidad y seguridad.

El proyecto comenzó hace más de un año (Junio del 2006) , con un diseño similar al propuesto en el presente documento. Se realizó una migración del antiguo sistema al nuevo, durante el mes de Julio del año pasado, y surgieron varios problemas que se fueron solucionando progresivamente. Este sistema en pruebas tenía cuatro máquinas físicas, dos de las cuales fallaron por problemas hardware, por lo que hubo que comprar una máquina nueva que sustituyese a las inoperativas.

La máquina nueva (*Donald*) se compró pensando en el rediseño que se haría del sistema que estaba funcionando en ese momento. Ese rediseño es el que aparece en la presente memoria, y que se ha implantado a lo largo del mes de mayo y junio del año en curso (2007).

El actual sistema es un sistema estable, basado en máquinas virtuales y autenticación por LDAP, y que el grupo ARCOS utiliza diariamente.

7.2.3. Aportaciones personales

La realización del proyecto me ha aportado valiosos conocimientos sobre el mundo de la virtualización y la administración bajo Linux. Una breve lista de las aportaciones que he obtenido con el proyecto es la siguiente:

- Conocimientos sobre los tipos de virtualización existentes.
- Manejo de máquinas virtuales en un sistema Linux.

- Administración de servidores en Linux.
- Aprendizaje de Ldap para autenticación.
- Aprendizaje de Samba como controlador de dominio utilizando Ldap.
- Instalación y configuración de distintos servicios bajo Linux.

También he de mencionar la experiencia de trabajo real que he obtenido con éste proyecto, dado que del fruto de mi trabajo dependía todo el personal de ARCOS directamente implicado con la implantación del nuevo sistema.
He aprendido a trabajar como consultor informático, atendiendo no solo a la infraestructura del antiguo sistema, sino también a los usuarios y las necesidades que tenían para el nuevo sistema. Gracias a todos ellos, y escuchando sus peticiones, he podido analizar qué es lo que necesitaban realmente, y realizar un trabajo lo más profesional posible de análisis, diseño e implantación, para ofrecérselo.

Espero que este proyecto no solo sirva para obtener el titulo que tanto trabajo me ha costado ganarme, sino que sirva de referencia para aquellas personas que quieran utilizar las ventajas de la virtualización y la autenticación bajo Ldap en materia de servidores. Y por supuesto, con este proyecto pretendo aportar mi granito de arena a la comunidad Linux y al software libre, que espero que ganen el pulso a las compañías que siguen realizando software de código cerrado. Es un paso más para un mundo más libre.

7.3. Trabajos futuros

El sistema de servidores implantado en ARCOS cumple sus funciones de manera óptima, no obstante, quedan tareas pendientes que solventen incidencias encontradas y a la vez que mejoren el sistema.

La tarea más importante a realizar es la correcta implantación de *Daisy*, dado que en un principio se diseño su sistema para albergar 3 discos duros únicamente y además contaba con una distribución Debian Sarge (la antigua distribución estable de Debian, sustituida por Etch). Actualmente se está trabajando en la reinstalación del sistema tal y como aparece en el anexo 2 (ver página 232.). Por motivo de la reinstalación, no se han podido obtener gráficas de su comportamiento para explicar en el capítulo de resultados.

El resto de tareas a realizar se se exponen a continuación:

- **Implantación de un sistema *firewall* mediante Iptables**: actualmente la seguridad de todo el sistema es elevada dado que ha sido uno de los ítems a tener en cuenta en el diseño. No obstante, la adicción de un sistema de *firewall* mediante

Iptables aumentaría la seguridad de todo el sistema, aislando de una forma más concisa cada parte del sistema.

- **Actualización automática de paquetes de seguridad**: en la actualidad, cada máquina que compone el sistema, tiene en su fichero */etc/apt/sources.list* (direcciones de los repositorios de paquetes en Debian), la dirección del repositorio de actualizaciones de seguridad. Esto se hace para actualizar en el sistema solo aquellos paquetes modificados o parcheados por temas de seguridad. Si se necesita instalar un paquete nuevo, se añadirán temporalmente al fichero *sources.list* el resto de repositorios, y una vez instalado el paquete, se deja de nuevo el repositorio de actualizaciones de seguridad únicamente.

- **Reubicación de ficheros *log* rotados**: en cada una de las máquinas virtuales está instalado el paquete *logrotate*, cuya función es rotar los ficheros *log* cuando éstos llegan a ocupar un tamaño determinado. Los ficheros *log* rotados que genera la herramienta, van ocupando demasiado espacio en la máquina, aunque se puede configurar la herramienta *logrotate* para que vaya borrando los ficheros rotados, manteniendo una ocupación constante y fijada por el administrador. No obstante, se desea en algunos casos mantener ficheros rotados de varios meses, con el inconveniente añadido de que ocupan un espacio importante en el sistema. Una solución es crear una cuenta especial de administración y copiar en ella todos los *log* rotados, dejando el directorio */var/log* de las máquinas con los ficheros que se estén utilizando únicamente.

- **Reubicación de los correos del usuario root**: en general, los correos que llegan a la cuenta del usuario *root* de cada máquina son ignorados. Esto sucede por no tener un mecanismo más automático de lectura de los mismos, obligando a los administradores a entrar en cada máquina y leer el correo con el comando *mail*, resultando ser una tarea tediosa. Una de las mejoras propuestas para el sistema es la de mover dichos correos a una cuenta especial de administración, facilitando su lectura a los administradores, que pueden acceder al correo de dicha cuenta por medio de un cliente de correo (como Thunderbird por ejemplo).

- **Generación de páginas web con los resultado de cada informe**: actualmente, existen varios *scripts* que generan informes sobre el estado del sistema, enviándose por correo a los administradores. Un ejemplo de estos *scripts* son los que se encargan de realizar las copias de seguridad, cuyo informe de transferencia es enviado por correo. Lo que se pretende es crear un interfaz web que reciba dichos informes y genere una página con los resultados de los mismos. La idea es que los administradores accedan a dicha web, en la que se utilice algún tipo de señalización que indique los estados correcto, incorrecto y alerta, de cada informe, y ellos solo tengan que revisar las actividades cuyo informe ha generado la señalización de incorrecto o

alerta. De esta forma, la revisión de los informes por parte de los administradores será una tarea menos laboriosa.

- **Implantación de un sistema de sincronización a través de red**: según el diseño especificado para éste sistema, la sincronización de datos entre Donald y Daisy se realiza mediante *rsync* una vez al día. Ante cualquier incidente que provoque la entrada en funcionamiento de Daisy en modo respaldo de Donald, la copia de los datos que contiene Daisy tendrá en el peor de los casos, un día de diferencia.

  Hay sistemas que comparten dispositivos físicos a través de la red, como Network Block Device (NBD). La idea es utilizar esta tecnología para compartir los volúmenes lógicos a través de la red, creando un RAID1 sobre ellos para que ambas máquinas tengan los mismos datos en sus volúmenes lógicos. La figura 7.1 ilustra esta operación. En el sistema actual se diseñaría de la siguiente forma:

  - Donald exportaría sus volúmenes lógicos mediante el demonio servidor NBD.
  - Daisy importaría estos volúmenes lógicos mediante el demonio cliente NBD.
  - Daisy crearía un RAID1 software con los volúmenes lógicos importados y los volúmenes lógicos locales creados en Daisy.

  La implementación de ésta tecnología permitiría un sistema de alta disponibilidad completo. En el caso de que fallase Donald, Daisy entraría en modo respaldo funcionando completamente y con todos los datos del sistema actualizados.

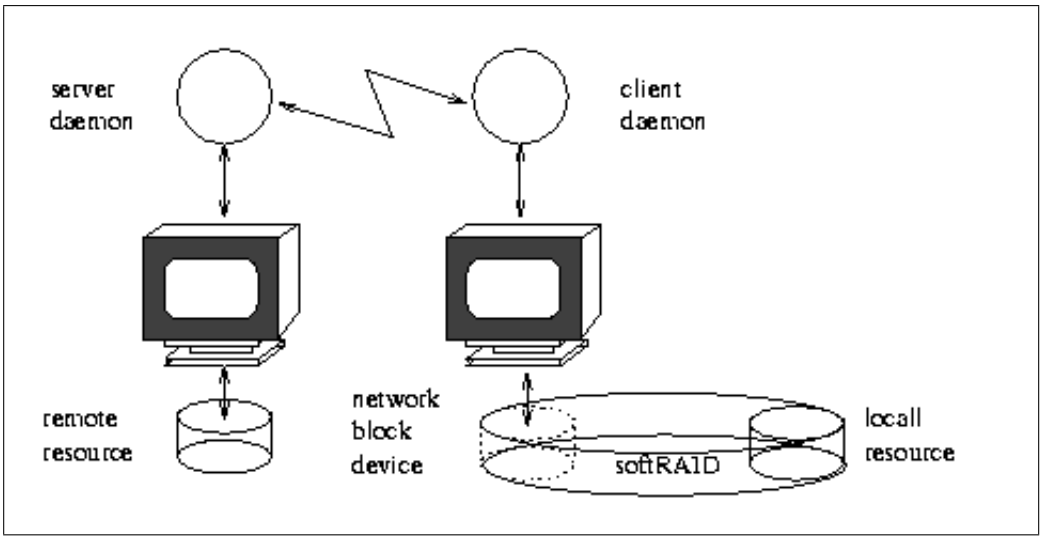


Figura 7.1: Creación de un RAID utilizando Network Block Device

Al margen de las mejoras o modificaciones a efectuar sobre el sistema, con objeto de mejorar la eficiencia del mismo, existe un proyecto futuro como ampliación del presente proyecto. La idea es automatizar al máximo la creación de cada máquina virtual, y crear un *script* que permita generar un sistema de servidores completo, similar al del presente documento. El *script* permitiría parametrizar al máximo el sistema según las necesidades que se tengan, y generaría cada máquina virtual, instalando los servicios seleccionados y configurándolos automáticamente para que ofrezcan la funcionalidad buscada. Se busca, por tanto, un generador de sistemas de servidores basados en máquinas virtuales y autenticación Ldap.

Capítulo 8

Apéndices

8.1. Anexo 1 - Instalación del servidor *Donald*

En este artículo se detalla la instalación y configuración del servidor *Donald*, destinado a servir de sistema anfitrión para máquinas virtuales Xen.

8.1.1. Instalación del sistema base con Knoppix

Para llevar a cabo la instalación del sistema base del servidor *Donald* se va a hacer uso de la herramienta *debootstrap* proporcionada por la distribución de Linux Debian. Ésta herramienta consiste en la descarga de un sistema base Linux preconfigurado para realizar sobre él las configuraciones básicas que permitan arrancar el sistema.

Se inicia la máquina con la distribución Knoppix versión 5.1, con las opciones: *lang=es 2 dma* ; configuración de consola en español, modo monousuario y *dma* activado para los discos.

Particionamiento

La máquina cuenta con 8 discos Serial ATA introducidos según aparece en el cuadro 8.1.

Se desea crear 3 volúmenes lógicos para los espacios de disco *usuarios backup_usuarios* y *sistema*. Para su creación se utilizará un solo grupo de volumen formado por 4 volúmenes físicos, previamente creados. Estos volúmenes físicos serán RAID de tipo 1 (*md0*, *md1*, *md2* y *md3*), construidos según muestra el cuadro 8.2.

Posición	**Modelo**	**S/N**	**Capacidad**	**Puerto**	**Dispositivo**
1º	SEAGATE ST3250823AS	5ND04QHE	250GB	4	*/dev/sde*
2º	SEAGATE ST3250823AS	5ND0TLWZ	250GB	5	*/dev/sdf*
3º	MAXTOR 7Y250M0	Y61PSCZE	250GB	6	*/dev/sdg*
4º	MAXTOR 6V250F0	V505TX5G	250GB	7	*/dev/sdh*
5º	MAXTOR 7Y250M0	Y6251GKE	250GB	3	*/dev/sdd*
6º	MAXTOR 7Y250M0	Y61R9PNE	250GB	2	*/dev/sdb*
7º	SEAGATE ST3400833AS	4NF16QWQ	400GB	1	*/dev/sdc*
8º	SEAGATE ST3400833AS	4NF1E3JL	400GB	0	*/dev/sda*

Cuadro 8.1: Colocación física de los discos duros en *Donald*

RAID	**Dispositivo**	**Puerto**
md0	sda3	0
	sdc3	1
md1	sdb1	2
	sdd1	3
md2	sde1	4
	sdf1	5
md3	sdg1	6
	sdh1	7
md4	sda1	0
	sdc1	1

Cuadro 8.2: Distribución de las particiones en varios RAID de tipo 1 en *Donald*

El último RAID *(md4)*, se utilizará para contener el sistema operativo y las máquinas virtuales.

Para la creación de los RAID se utiliza el comando *mdadm*:

```
# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda3 /dev/sdc3
# mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdd1
# mdadm --create /dev/md2 --level=1 --raid-devices=2 /dev/sde1 /dev/sdf1
# mdadm --create /dev/md3 --level=1 --raid-devices=2 /dev/sdg1 /dev/sdh1
# mdadm --create /dev/md4 --level=1 --raid-devices=2 /dev/sda1 /dev/sdc1
```

A continuación se crean los volúmenes físicos:

```
# pvcreate /dev/md0
# pvcreate /dev/md1
# pvcreate /dev/md2
# pvcreate /dev/md3
```

Una vez creados es necesario crear un grupo de volúmen que agrupe los volúmenes físicos creados anteriormente:

```
# vgcreate vg1 /dev/md0 /dev/md1 /dev/md2 /dev/md3
```

Y por último se crean los 3 volúmenes lógicos utilizando todo el espacio disponible en el grupo de volumen:

```
# lvcreate -L 300G -n sistema vg1
# lvcreate -L 361,7G -n backup_usuarios vg1
# lvcreate -L 361,7G -n usuarios vg1
```

Una vez creados todos los dispositivos necesarios, ya se pueden formatear para crear un sistema de ficheros por encima. En este caso, se selecciona ext3:

```
# mkfs.ext3 /dev/vg1/sistema
# mkfs.ext3 /dev/vg1/backup_usuarios
# mkfs.ext3 /dev/vg1/usuarios
# mkfs.ext3 /dev/md4
# mkswap /dev/sda2
# mkswap /dev/sdc2
```

Descarga del sistema base

Se monta el RAID */dev/md4* que contendrá el sistema base en */mnt/raiz*, y se ejecuta el siguiente comando para descargar el sistema base del repositorio indicado:

```
# debootstrap --arch i386 etch /mnt/raiz ftp://ftp.rediris.es/debian
```

Para la configuración del sistema base es necesario introducirse en el mismo como si del sistema anfitrión se tratase. El comando *chroot* proporciona dicha utilidad, cambiando el directorio raíz del sistema al indicado y ejecutando una *shell* con el nuevo directorio *root*. Previamente a la ejecución del comando, es importante montar el sistema de ficheros virtual *proc* dentro del propio *chroot*. Además, Knoppix monta los discos con las opciones *nosuid,nodev*, siendo recomendable cambiarlas por *suid* y *dev* para evitar futuros problemas en la instalación de paquetes que requieran acceso a dispositivos.

```
# mount -o remount,suid,dev /mnt/raiz
# mount -t proc proc /mnt/raiz/proc
# chroot /mnt/raiz
```

Entrada al entorno *chroot*:

Una vez dentro del *chroot* es posible ejecutar ciertos comandos sin comprometer el sistema anfitrión. Además proporciona una mayor comodidad en la configuración de los ficheros del sistema descargado, ya que se evita el fallo común de confundirlo con el sistema anfitrión.

Es necesario configurar el *apt* con las fuentes (*sources*) de la rama estable de Debian, que en este momento se trata de Debian Etch. Se edita el fichero */etc/apt/sources.list* y se añaden las fuentes (ver página 229.).

A continuación se actualiza la lista de paquetes:

```
# apt-get update
```

Y después se han de descargar los siguientes paquetes:

- ***localeconf***: configura las variables del sistema para utilizar el conjunto de caracteres seleccionado segun el idioma.
- ***vim***: instala el editor Vim Improved, más cómodo de utilizar.
- ***less***: herramienta similar al comando *more*.
- ***grub***: gestor de arranque necesario para que inicie la máquina.
- ***locales***: conjunto de ficheros y herramientas que permiten la localización e internacionalización de programas que utilizan las librerías de C.
- ***console-data***: conjunto de paquetes que incluye definiciones del mapa de teclado, fuentes para la consola y varias codificaciones del conjunto de caracteres a utilizar en el sistema.
- ***xen-linux-system-2.6.18-4-xen-vserver-686***: paquete que recoge las utilidades necesarias para ejecutar y monitorizar máquinas Xen.
- ***linux-image-2.6.18-4-xen-vserver-686***: *kernel* Xen 2.6.18-4 tanto para el sistema anfitrión como para las máquinas virtuales.
- ***linux-modules-2.6.18-4-xen-vserver-686***: módulos para el *kernel* Xen 2.6.18-4.
- ***xen-hypervisor-3.0.3-1-i386-pae***: Xen hypervisor separa el hardware físico del sistema operativo.
- ***memtest86***: Utilidad para realizar test de memoria.

- ***bridge-utils***: Utilidades para la creación de puentes (*bridges*) de red.
- ***mdadm***: Utilidad para la gestión de dispositivos RAID en Linux.
- ***lvm2***: Utilidad para la gestión de volúmenes lógicos en Linux.

A continuación se debe personalizar el sistema mediante varias acciones:

- **Nombre del *host*:** la maquina debe ser nombrada y configurada para actuar como servidor. Mediante el siguiente comando, se le indica el nombre a utilizar:

```
# hostname donald
```

- **Interfaces de red:** la IP asignada para *Donald* es la **163.117.148.242**, se modifica el fichero */etc/network/interfaces* para que utilice dicha ip (ver página 229.).
- **DNS:** *Donald* ha de utilizar a **163.117.131.31** (DNS de la Universidad) temporalmente como servidor DNS. Esto se configura en el fichero */etc/resolv.conf* (ver página 229.).
- ***Hosts*:** se edita el fichero */etc/hosts* indicando cual es el *localhost* (ver página 229.).
- ***Fstab*:** se edita el fichero */etc/fstab* para que utilice */dev/md4* como raíz del sistema, *sda2* y *sdc2* como *swap*, y se especifican directorios en */mnt* para automatizar el montaje de los dispositivos existentes (ver página 229.).
- **Contraseña:** a continuación se introduce la contraseña de *root*:

```
# passwd
Enter new UNIX password: ******
Retype new UNIX password: *****
passwd: password updated successfully
```

- ***Locales*:** para poder definir y utilizar valores de localización distintos del inglés, se utiliza el paquete *locales*, en cuya instalación se han seleccionado las siguientes *locales* a utilizar:

```
es_ES@euro ISO-8859-15
es_ES ISO-8859-1
es_ES.UTF-8@euro UTF-8
es_ES.UTF-8 UTF-8
```

Se ha de seleccionar, para utilizar en el sistema como *locale* por defecto, la *'es_ES.UTF-8@euro UTF-8'*

- **Mapa del teclado:** para poder utilizar el teclado en español se utiliza el paquete console-data, en el cual se selecciona la siguiente configuración:

  ```
  pc / qwerty / Spanish / Standard / Standard
  ```

- **Configuración de Xen:** es necesario definir en el fichero */etc/xen/xend-config.sxp* que se utilizará el modo *bridge* para la red de las máquinas virtuales, y que el *dom0* (sistema *host*) utilizará un mínimo de 1Gb de memoria (ver página 231.).

Instalación del *kernel*

Se ha comprobado al intentar iniciar la máquina *Donald* con versiones de *kernel* inferiores a la 2.6.19 que el *driver* de la controladora Serial Ata (*3w_9xxx*) no detecta los discos duros. A partir de la versión 2.6.19 del *kernel* se ha implementado una mejora en dicho driver [1], permitiendo detectar y utilizar los discos duros.

Como es necesaria la utilización de un *kernel* con soporte para Xen, y Debian incluye parches para los *kernel* de la rama 2.6.18 que incorporan el soporte para Xen, es necesario buscar una solución para poder utilizar el *driver* mejorado de la rama 2.6.19 y un *kernel* 2.6.18 parcheado para Xen.

La solución buscada ha consistido en utilizar un *kernel* 2.6.18 con el parche de Debian para Xen y con el *driver* que incluye la versión del *kernel* 2.6.19 del controlador *3w_9xxx*, sobreescribiendo la versión que trae el propio *kernel* 2.6.18. [11]

Los pasos han sido los siguientes (como usuario *root*):

- Descarga de los paquetes necesarios para la instalación de un *kernel*:

  ```
  apt-get install gcc make patch libncurses5-dev
  ```

- Acceso al directorio donde se ha de descargar y descomprimir las fuentes del *kernel* y descarga de las mismas:

  ```
  cd /usr/src/
  wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.18.tar.bz2
  tar -xjf linux-2.6.18.tar.bz2
  ```

- Descarga del parche oficial de la versión de *kernel* 2.6.18 para convertirlo en la versión 2.6.18.8:

```
wget http://www.kernel.org/pub/linux/kernel/v2.6/patch-2.6.18.8.bz2
bunzip2 patch-2.6.18.8.bz2
```

- Descarga del paquete de parches aplicados al *kernel* de Debian, que incluye el parche de Xen:

```
apt-get install linux-patch-debian-2.6.18
```

- Renombrado del directorio del núcleo y parcheado del mismo:

```
mv linux-2.6.18 linux-2.6.18.8-xen
mv patch-2.6.18.8 linux-2.6.18.8-xen/
bunzip2 /usr/src/kernel-patches/all/2.6.18/debian/features/all/xen/fedora-2.6.18-36186.patch.bz2
cp /usr/src/kernel-patches/all/2.6.18/debian/features/all/xen/fedora-2.6.18-36186.patch \
  linux-2.6.18.8-xen/
bunzip2 /usr/src/kernel-patches/all/2.6.18/debian/bugfix/all/xen/swiotlb-highmem-copy.patch.bz2
cp /usr/src/kernel-patches/all/2.6.18/debian/bugfix/all/xen/swiotlb-highmem-copy.patch \
  linux-2.6.18.8-xen/
cd linux-2.6.18.8-xen/
patch -s -p1 < patch-2.6.18.8
patch -s -p1 < fedora-2.6.18-36186.patch
patch -s -p1 < swiotlb-highmem-copy.patch
```

- Vuelta al directorio anterior y descarga de la versión del *kernel* 2.6.19:

```
cd /usr/src/
wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.19.tar.bz2
tar -xjf linux-2.6.19.tar.bz2
```

- Sobreescritura del *driver 3w_9xxx* de la versión de *kernel* 2.6.19 sobre la versión 2.6.18.8, realizando previamente una copia de seguridad.

```
mkdir /usr/src/linux-2.6.18.8-xen/drivers/scsi/backup
cp /usr/src/linux-2.6.18.8-xen/drivers/scsi/3w* \
   /usr/src/linux-2.6.18.8-xen/drivers/scsi/backup/.
cp -f /usr/src/linux-2.6.19/drivers/scsi/3w* \
   /usr/src/linux-2.6.18.8-xen/drivers/scsi/.
```

- Creación de un enlace simbólico para definir el *kernel* por defecto del sistema, configuración del fichero *Makefile* para que utilice el nombre requerido y configuración del *kernel*:

```
ln -s /usr/src/linux-2.6.18.8-xen/ /usr/src/linux
cd /usr/src/linux
vi Makefile ( la línea 4 ha de tener: EXTRAVERSION = .8-xen)
make menuconfig
```

Al menos deberán seleccionarse las siguientes opciones:

```
Processor type and features ---> Subarchitecture Type (Xen-compatible)
Xen --->
  Privileged Guest (domain 0)
  Backend driver support
    Block-device backend driver
    Network-device backend driver
      Network-device loopback driver
  PCI device backend driver
    PCI Backend Mode (Virtual PCI)
  Block-device frontend driver
  Network-device frontend driver
  Scrub memory before freeing it to Xen
  Disable serial port drivers
  Export Xen attributes in sysfs
  Xen version compatibility (3.0.2 and later)

Networking --->
  Networking Options --->
    802.1d Ethernet Bridging
```

- Compilación del *kernel*:

```
make && make modules_install
```

- Una vez terminada la compilación, se instala el *kernel* y se crea la imagen *initrd*:

```
cp vmlinuz /boot/vmlinuz-2.6.18.8-xen
cp System.map /boot/System.map-2.6.18.8-xen
update-initramfs -k 2.6.18.8-xen -c
```

- Finalmente, hay que definir en el fichero */etc/modules* los módulos a cargar en el sistema en el arranque. Ver 8.1.3 en la página 230.

Salida del entorno *chroot*:

Se necesita abandonar el entorno *chroot* para poder crear, desde Knoppix el sector de arranque en */dev/sda1*. Para ello se debe teclear:

```
# exit
```

Retornando a la *shell* de Knoppix.

A continuación, hay que indicar al *grub* que instale el sector de arranque

```
# grub-install --root-directory=/mnt/raiz --no-floppy /dev/sda1
```

Por último, la máquina ha de iniciar como sistema anfitrión y con el kernel *2.6.18.8-xen*. Se ha de indicar en el fichero */mnt/raiz/boot/grub/menu.lst* el *kernel* a arracar y los parámetros de arranque (ver página 230.).

8.1.2. Reinicio del sistema y prueba del nuevo *kernel*

El sistema esta listo para ser iniciado, se teclea:

```
# reboot
```

y se retira el CD de Knoppix. La maquina reiniciará por tanto con el nuevo *kernel 2.6.18.8-xen*; se ha de comprobar que arranca de una manera correcta y esperar a que aparezca la pantalla de *login*:

```
Debian GNU/Linux 4.0 donald tty1


donald login:
```

Se introduce *login* y *password* y se comprueba que los interfaces de red están funcionando:

```
# ifconfig
```

Han de aparecer los interfaces, *eth0*, *lo* y *xen-br0*.

Se ha de comprobar también que el *Dom 0* (*sistema anfitrión*) esta ejecutándose:

```
# xm list
```

ha de aparecer:

```
Name                                        ID Mem(MiB) VCPUs State   Time(s)
Domain-0                                     0     3922     2 r-----   5093.8
```

La máquina *Donald* estará en este punto perfectamente operativa.

8.1.3. Ficheros de configuración

Apt: */etc/apt/sources.list*

```
#Etch
deb ftp://ftp.de.debian.org/debian/ etch main non-free contrib
deb-src ftp://ftp.de.debian.org/debian/ etch  main non-free contrib
deb http://security.debian.org/ etch/updates main contrib non-free
deb http://ftp.rediris.es/debian/ etch main contrib non-free
```

Interfaces de red: */etc/network/interfaces*

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# This entry denotes the loopback (127.0.0.1) interface.
auto lo
iface lo inet loopback

# This entry was created during the Debian installation
# (network, broadcast and gateway are optional)
auto eth0
iface eth0 inet static
        address 163.117.148.242
        netmask 255.255.255.0
        network 163.117.148.0
        broadcast 163.117.148.255
        gateway 163.117.148.2
```

DNS: */etc/resolv.conf*

```
search arcos.inf.uc3m.es
nameserver 163.117.131.31
```

Fstab: */etc/fstab*

```
/dev/md4        /               ext3    defaults        0       1
proc            /proc           proc    defaults        0       0
/dev/sda2       swap            swap    defaults        0       0
/dev/sdc2       swap            swap    defaults        0       0
/dev/vg1/usuarios /mnt/usuarios ext3    noexec,nodev    0       2
/dev/vg1/backup_usuarios /mnt/backup_usuarios ext3 noexec,nodev 0 3
/dev/vg1/sistema  /mnt/sistema  ext3    noexec,nodev    0       4
sysfs           /sys            sysfs   defaults        0       0
tmpfs           /dev/shm        tmpfs   defaults        0       0
```

Hosts: */etc/hosts*

```
127.0.0.1       localhost
163.117.148.242 donald
```

Módulos: */etc/modules*

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
loop
netloop
ahci
3w_9xxx
shpchp
sbs
ac
bridge
```

Grub: */boot/grub/menu.lst*

```
#
# Sample boot menu configuration file
#

# Boot automatically after 30 secs.
timeout 8

# By default, boot the first entry.
default 0

# Fallback to the second entry.
fallback 1


title  GNU/Linux 2.6.18.8-xen
root (hd0,0)
kernel /boot/xen-3.0.3-1-i386-pae.gz noreboot
module /boot/vmlinuz-2.6.18.8-xen root=/dev/md4 console=tty0
module /boot/initrd.img-2.6.18.8-xen

title memtest86
root (hd0,1)
kernel /boot/memtest86.bin


# For installing GRUB into the hard disk
title Install GRUB into the hard disk
root    (hd0,1)
setup   (hd0)

# Change the colors.
title Change the colors
```

```
color light-green/brown blink-red/bluesavedefault
```

Configuración de Xen: */etc/xen/xend-config.sxp*

```
(network-script 'network-bridge netdev=eth0')
(vif-script vif-bridge)
(dom0-min-mem 1024)
(dom0-cpus 0)
```

8.2. Anexo 2 - Instalación del servidor *Daisy*

En este artículo se detalla la instalación y configuración del servidor *Daisy*, destinado a servir de sistema anfitrión para máquinas virtuales Xen, como respaldo de *Donald*.

8.2.1. Instalación del sistema base con Knoppix

Para llevar a cabo la instalación del sistema base del servidor *Daisy* se va a hacer uso de la herramienta *debootstrap* proporcionada por la distribución de Linux Debian. Ésta herramienta consiste en la descarga de un sistema base Linux preconfigurado para realizar sobre él las configuraciones básicas que permitan arrancar el sistema.

Se inicia la máquina con la distribución Knoppix versión 5.1, con las opciones: *lang=es 2 dma* ; configuración de consola en español, modo monousuario y *dma* activado para los discos.

Particionamiento

La máquina cuenta con 4 discos ATA 100 con las características que aparecen en el cuadro 8.3.

Modelo	**S/N**	**Capacidad**	**Dispositivo**
SEAGATE ST3250820ACE	9QE12XHZ	250GB	*/dev/hda*
MAXTOR 7Y250P0	Y60X58FE	250GB	*/dev/hdb*
SEAGATE ST3250824A	5ND3TL9R	250GB	*/dev/hdc*
SEAGATE ST3250820A	9QE4XGP8	250GB	*/dev/hdd*

Cuadro 8.3: Información de los discos duros en *Daisy*

Se desea crear un RAID5 para almacenar los volúmenes lógicos que servirán de respaldo de los que contiene *Donald*. Es decir, los espacios de disco *usuarios backup_usuarios* y *sistema*. Para su creación se utilizarán las particiones *hda3*, *hdb3*, *hdc3* y *hdd3*. El RAID5 creado será finalmente el dispositivo */dev/md1*.

También se desea crear un RAID10 para almacenar el sistema raíz de *Daisy*, para ello se utilizarán las particiones */dev/hda1*, */dev/hdb1*, */dev/hc1* y */dev/hdd1*. El RAID1 creado será finalmente el dispositivo */dev/md0*.

Como el sistema raiz estará almacenado en un RAID10, no se podrá iniciar desde él. La solución es crear una partición para el directorio */boot*. Dado que se tienen 4 discos

duros con el mismo particionamiento, se aprovecharán las 4 particiones de los 4 discos duros para crear un RAID1 con dos de ellas y dejar las 2 restantes como *spare disks* del RAID1 creado. Este RAID1 será el dispositivo */dev/md2* y las particiones del RAID1 serán: */dev/hda4* y */dev/hdc4*. Por otro lado, las particiones spare disk del RAID1 serán: */dev/hdb4* y */dev/hdd4*.

El resto de particiones (*/dev/hda2*, */dev/hdb2*, */dev/hc2* y */dev/hdd2*) se utilizarán como *swap* del sistema. El cuadro 8.4 refleja el particionamiento a realizar en *Daisy*.

Partición	**Dispositivo**
hda1 hdb1 hdc1 hdd1	*/dev/md0*
hda2 hdb2 hdc2 hdd2	*swap*
hda3 hdb3 hdc3 hdd3	*/dev/md1*
hda4 hdb4 hdc4 hdd4	*/dev/md2*

Cuadro 8.4: Distribución de las particiones en *Daisy*

Para la creación de los RAID, tanto */dev/md0* como */dev/md1*, se utiliza el comando *mdadm*:

```
# mdadm --create /dev/md0 --level=10 --raid-devices=4 /dev/hda1 \
                        /dev/hdb1 /dev/hdc1 /dev/hdd1
# mdadm --create /dev/md1 --level=5 --raid-devices=4 /dev/hda3 \
                        /dev/hdb3 /dev/hdc3 /dev/hdd3
# mdadm --create /dev/md2 --level=1 --raid-devices=2 \
       --spare-devices=2 /dev/hda4 /dev/hdc4 /dev/hdb4 /dev/hdd4
```

A continuación se crean los volúmenes físicos:

```
# pvcreate /dev/md1
```

Una vez creados es necesario crear un grupo de volumen que agrupe los volúmenes físicos creados anteriormente:

```
# vgcreate vg1 /dev/md1
```

Y por último se crean los 3 volúmenes lógicos utilizando todo el espacio disponible en el grupo de volumen:

```
# lvcreate -L 250G -n sistema vg1
# lvcreate -L 200G -n usuarios vg1
# lvcreate -L 175G -n backup_usuarios vg1
```

Una vez creados todos los dispositivos necesarios, ya se pueden formatear para crear un sistema de ficheros por encima. En este caso, se selecciona *ext3*:

```
# mkfs.ext3 /dev/vg1/sistema
# mkfs.ext3 /dev/vg1/backup_usuarios
# mkfs.ext3 /dev/vg1/usuarios
# mkfs.ext3 /dev/md0
# mkfs.ext3 /dev/md2
# mkswap /dev/hda2
# mkswap /dev/hdb2
# mkswap /dev/hdc2
# mkswap /dev/hdd2
```

Descarga del sistema base

Se monta el RAID */dev/md0* que contendrá el sistema base en */mnt/raiz*, y se ejecuta el siguiente comando para descargar el sistema base del repositorio indicado:

```
# debootstrap --arch i386 etch /mnt/raiz ftp://ftp.rediris.es/debian
```

Para la configuración del sistema base es necesario introducirse en el mismo como si del sistema anfitrión se tratase. El comando *chroot* proporciona dicha utilidad, cambiando el directorio raíz del sistema al indicado y ejecutando una *shell* con el nuevo directorio *root*. Previamente a la ejecución del comando, es importante montar el sistema de ficheros virtual *proc* dentro del propio *chroot*. Además, Knoppix monta los discos con las opciones *nosuid,nodev*, siendo recomendable cambiarlas por *suid* y *dev* para evitar futuros problemas en la instalación de paquetes que requieran acceso a dispositivos.

```
# mount -o remount,suid,dev /mnt/raiz
# mount -t proc proc /mnt/raiz/proc
# chroot /mnt/raiz
```

Entrada al entorno *chroot*:

Una vez dentro del *chroot* es posible ejecutar ciertos comandos sin comprometer el sistema anfitrión. Además proporciona una mayor comodidad en la configuración de los ficheros del sistema descargado, ya que se evita el fallo común de confundirlo con el sistema anfitrión.

Es necesario configurar el *apt* con las fuentes (*sources*) de la rama estable de Debian, que en este momento se trata de Debian Etch. Se edita el fichero */etc/apt/sources.list* y se añaden las fuentes (ver página 238.).

A continuación se actualiza la lista de paquetes:

```
# apt-get update
```

Y después se han de descargar los siguientes paquetes:

- ***localeconf***: configura las variables del sistema para utilizar el conjunto de caracteres seleccionado segun el idioma.
- ***vim***: instala el editor Vim Improved, más cómodo de utilizar.
- ***less***: herramienta similar al comando *more*.
- ***grub***: gestor de arranque necesario para que inicie la máquina.
- ***locales***: conjunto de ficheros y herramientas que permiten la localización e internacionalización de programas que utilizan las librerías de C.
- ***console-data***: conjunto de paquetes que incluye definiciones del mapa de teclado, fuentes para la consola y varias codificaciones del conjunto de caracteres a utilizar en el sistema.
- ***xen-linux-system-2.6.18-4-xen-vserver-686***: paquete que recoge las utilidades necesarias para ejecutar y monitorizar máquinas Xen.
- ***linux-image-2.6.18-4-xen-vserver-686***: *kernel* Xen 2.6.18-4 tanto para el sistema anfitrión como para las máquinas virtuales.
- ***linux-modules-2.6.18-4-xen-vserver-686***: módulos para el *kernel* Xen 2.6.18-4.
- ***xen-hypervisor-3.0.3-1-i386-pae***: Xen hypervisor separa el hardware físico del sistema operativo.
- ***memtest86***: Utilidad para realizar test de memoria.

- ***bridge-utils***: Utilidades para la creación de puentes (*bridges*) de red.
- ***mdadm***: Utilidad para la gestión de dispositivos RAID en Linux.
- ***lvm2***: Utilidad para la gestión de volúmenes lógicos en Linux.

A continuación se debe personalizar el sistema mediante varias acciones:

- **Nombre del *host*:** la maquina debe ser nombrada y configurada para actuar como servidor. Mediante el siguiente comando, se le indica el nombre a utilizar:

```
# hostname daisy
```

- **Interfaces de red:** la IP asignada para *Daisy* es la **163.117.148.243**, se modifica el fichero */etc/network/interfaces* para que utilice dicha IP (ver página 238.).
- **DNS:** *Daisy* ha de utilizar a **163.117.131.31** (DNS de la Universidad) temporalmente como servidor DNS. Esto se configura en el fichero */etc/resolv.conf* (ver página 239.).
- ***Hosts*:** se edita el fichero */etc/hosts* indicando cual es el *localhost* (ver página 239.).
- ***Fstab*:** se edita el fichero */etc/fstab* para que utilice */dev/md0* como raíz del sistema, *hda2*, *hdb2*, *hdc2* y *hdd2* como *swap*, y se especifican directorios en */mnt* para automatizar el montaje de los dispositivos existentes (ver página 239.).
- **Contraseña:** a continuación se introduce la contraseña de *root*:

```
# passwd
Enter new UNIX password: ******
Retype new UNIX password: *****
passwd: password updated successfully
```

- ***Locales*:** para poder definir y utilizar valores de localización distintos del inglés, se utiliza el paquete *locales*, en cuya instalación se han seleccionado las siguientes *locales* a utilizar:

```
es_ES@euro ISO-8859-15
es_ES ISO-8859-1
es_ES.UTF-8@euro UTF-8
es_ES.UTF-8 UTF-8
```

Se ha de seleccionar, para utilizar en el sistema como *locale* por defecto, la *'es_ES.UTF-8@euro UTF-8'*

- **Mapa del teclado:** para poder utilizar el teclado en español se utiliza el paquete console-data, en el cual se selecciona la siguiente configuración:

  `pc / qwerty / Spanish / Standard / Standard`

- **Configuración de Xen:** es necesario definir en el fichero */etc/xen/xend-config.sxp* que se utilizará el modo *bridge* para la red de las máquinas virtuales, y que el *dom0* (sistema *host*) utilizará un mínimo de 196Mb de memoria (ver página 240.).

Salida del entorno *chroot*:

Se necesita abandonar el entorno *chroot* para poder crear, desde Knoppix el sector de arranque en */dev/hda*. Para ello se debe teclear:

```
# exit
```

Retornando a la *shell* de Knoppix.

A continuación, hay que indicar al *grub* que instale el sector de arranque. Previamente, montamos el RAID1 que contiene el directorio */boot* de *Daisy*:

```
# mkdir /mnt/md2
# mount /dev/md2 /mnt/md2
# cd /mnt/md2
# ln -s . boot ( este enlace simbólico se crea para engañar al grub-install,
                 dado que busca el directorio /boot y ese directorio es la
                 propia partición montada )
# grub-install --root-directory=/mnt/md2 --no-floppy /dev/hda
```

Por último, la máquina ha de iniciar como sistema anfitrión y con el *kernel 2.6.18.8-xen*. Se ha de indicar en el fichero */mnt/md2/grub/menu.lst* el *kernel* a arracar y los parámetros de arranque (ver página 240.).

8.2.2. Reinicio del sistema y prueba del nuevo *kernel*

El sistema esta listo para ser iniciado, se teclea:

```
# reboot
```

y se retira el CD de Knoppix. La maquina reiniciará por tanto con el nuevo *kernel 2.6.18.8-xen*; se ha de comprobar que arranca de una manera correcta y esperar a que aparezca la pantalla de *login*:

```
Debian GNU/Linux 4.0 daisy tty1


daisy login:
```

Se introduce *login* y *password* y se comprueba que los interfaces de red están funcionando:

```
# ifconfig
```

Han de aparecer los interfaces, *eth0*, *lo* y *xen-br0*.

Se ha de comprobar también que el *Dom 0* (*sistema anfitrión*) esta ejecutándose:

```
# xm list
```

ha de aparecer:

```
Name                                        ID Mem(MiB) VCPUs State   Time(s)
Domain-0                                     0      196     1 r-----     12.1
```

La máquina *Daisy* estará en este punto perfectamente operativa.

8.2.3. Ficheros de configuración

Apt: */etc/apt/sources.list*

```
#Etch
deb ftp://ftp.de.debian.org/debian/ etch main non-free contrib
deb-src ftp://ftp.de.debian.org/debian/ etch  main non-free contrib
deb http://security.debian.org/ etch/updates main contrib non-free
deb http://ftp.rediris.es/debian/ etch main contrib non-free
```

Interfaces de red: */etc/network/interfaces*

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# This entry denotes the loopback (127.0.0.1) interface.
auto lo
iface lo inet loopback

# This entry was created during the Debian installation
# (network, broadcast and gateway are optional)
auto eth0
iface eth0 inet static
        address 163.117.148.243
        netmask 255.255.255.0
        network 163.117.148.0
        broadcast 163.117.148.255
        gateway 163.117.148.2
```

DNS: */etc/resolv.conf*

```
search arcos.inf.uc3m.es
nameserver 163.117.131.31
```

Fstab: */etc/fstab*

```
/dev/md0        /               ext3    defaults        0       1
proc            /proc           proc    defaults        0       0
/dev/hda2       swap            swap    defaults        0       0
/dev/hdb2       swap            swap    defaults        0       0
/dev/hdc2       swap            swap    defaults        0       0
/dev/hdd2       swap            swap    defaults        0       0
/dev/md0        /mnt/md0        ext3    noauto,defaults 0       0
/dev/md2        /mnt/md2        ext3    noauto,defaults 0       0
/dev/md2        /boot           ext3    noauto,defaults 0       2
sysfs           /sys            sysfs   defaults        0       0
tmpfs           /dev/shm        tmpfs   defaults        0       0
```

Hosts: */etc/hosts*

```
127.0.0.1       localhost
163.117.148.243 daisy
```

Módulos: */etc/modules*

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

e1000
raid0
raid10
raid1
raid456
bridge
loop
netloop

# Generated by sensors-detect on Thu Sep 20 20:15:28 2007
# I2C adapter drivers
i2c-i810
i2c-i801
# Chip drivers
eeprom
w83627hf
```

Grub: */boot/grub/menu.lst*

```
# menu.lst - See: grub(8), info grub, update-grub(8)
#            grub-install(8), grub-floppy(8),
#            grub-md5-crypt, /usr/share/doc/grub
#            and /usr/share/doc/grub-doc/.

default         0

timeout         5

color cyan/blue white/blue

## ## End Default Options ##

title           Xen 3.0.3-1-i386-pae / Debian GNU/Linux, kernel 2.6.18-4-xen-vserver-686
root            (hd0,3)
kernel          /boot/xen-3.0.3-1-i386-pae.gz dom0_mem=196M
module          /boot/vmlinuz-2.6.18-4-xen-vserver-686 root=/dev/md0 ro console=tty0
module          /boot/initrd.img-2.6.18-4-xen-vserver-686
savedefault

title           Debian GNU/Linux, kernel memtest86
root            (hd0,3)
kernel          /boot/memtest86.bin

### END DEBIAN AUTOMAGIC KERNELS LIST
```

Configuración de Xen: */etc/xen/xend-config.sxp*

```
(network-script 'network-bridge netdev=eth0')

(vif-script vif-bridge)

(dom0-min-mem 196)

(dom0-cpus 0)
```

8.3. Anexo 3 - Instalación del servidor *Boyerito*

En este artículo se detalla la instalación y configuración del servidor *Boyerito*, destinado a servir de sistema de almacenamiento de las copias de seguridad realizadas a los volúmenes lógicos de *Donald* y *Daisy*.

8.3.1. Instalación del sistema base con Knoppix

Para llevar a cabo la instalación del sistema base del servidor *Boyerito* se va a hacer uso de la herramienta *debootstrap* proporcionada por la distribución de Linux Debian. Ésta herramienta consiste en la descarga de un sistema base Linux preconfigurado para realizar sobre él las configuraciones básicas que permitan arrancar el sistema.

Se inicia la máquina con la distribución Knoppix versión 5.1, con las opciones: *lang=es 2 dma* ; configuración de consola en español, modo monousuario y *dma* activado para los discos.

Particionamiento

La máquina cuenta con 4 discos IDE ATA 100 con las características que aparecen en el cuadro 8.5.

Modelo	**S/N**	**Capacidad**	**Dispositivo**
SEAGATE ST3120022A	3JS1HNXR	120GB	*/dev/hda*
SEAGATE ST3120022A	3JS1FW88	120GB	*/dev/hdb*
SEAGATE ST3120022A	3JS1HALF	120GB	*/dev/hdc*
SEAGATE ST3120022A	3JS1HNR0	120GB	*/dev/hdd*

Cuadro 8.5: Información de los discos duros en *Boyerito*

Se desea crear un RAID5 para almacenar la copia de seguridad de los volúmenes lógicos que utilizan *Donald* y *Daisy*, es decir, los espacios de disco *usuarios backup_usuarios* y *sistema*. Para su creación se utilizarán las particiones *hda3*, *hdb3*, *hdc3* y *hdd3*. El RAID5 creado será finalmente el dispositivo */dev/md0*.

También se desea crear un RAID1 para almacenar el sistema raíz de *Boyerito*, para ello se utilizarán las particiones */dev/hda2* y */dev/hdc2*. El RAID1 creado será finalmente el dispositivo */dev/md1*.

Partición	**Dispositivo**
hda1 hdc1	*swap*
hda2 hdc2	*/dev/md1*
hda3 hdb3 hdc3 hdd3	*/dev/md0*

Cuadro 8.6: Distribución de las particiones en *Boyerito*

El resto de particiones (*/dev/hda1* y */dev/hda2*) se utilizarán como *swap* del sistema. El cuadro 8.6 refleja el particionamiento a realizar en *Boyerito*.

Para la creación de los RAID, tanto */dev/md0* como */dev/md1*, se utiliza el comando *mdadm*:

```
# mdadm --create /dev/md0 --level=5 --raid-devices=4 /dev/hda3 \
          /dev/hdb3 /dev/hdc3 /dev/hdd3
# mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/hda2 /dev/hdc2
```

Una vez creados los dos RAID, ya se pueden formatear para crear un sistema de ficheros por encima. Además, también se pueden formatear las particiones destinadas a *swap*. En este caso, se selecciona *ext3* para los sistemas de ficheros:

```
# mkfs.ext3 /dev/md0
# mkfs.ext3 /dev/md1
# mkswap /dev/hda1
# mkswap /dev/hdc1
```

Descarga del sistema base

Se monta el RAID */dev/md1* que contendrá el sistema base en */mnt/raiz*, y se ejecuta el siguiente comando para descargar el sistema base del repositorio indicado:

```
# debootstrap --arch i386 etch /mnt/raiz ftp://ftp.rediris.es/debian
```

Para la configuración del sistema base es necesario introducirse en el mismo como si del sistema anfitrión se tratase. El comando *chroot* proporciona dicha utilidad, cambiando el directorio raíz del sistema al indicado y ejecutando una *shell* con el nuevo directorio *root*.

Previamente a la ejecución del comando, es importante montar el sistema de ficheros virtual *proc* dentro del propio *chroot*. Además, Knoppix monta los discos con las opciones *nosuid,nodev*, siendo recomendable cambiarlas por *suid* y *dev*, para evitar futuros problemas en la instalación de paquetes que requieran acceso a dispositivos.

```
# mount -o remount,suid,dev /mnt/raiz
# mount -t proc proc /mnt/raiz/proc
# chroot /mnt/raiz
```

Dentro del entorno *chroot*

Una vez dentro del *chroot* es posible ejecutar ciertos comandos sin comprometer el sistema anfitrión. Además proporciona una mayor comodidad en la configuración de los ficheros del sistema descargado, ya que se evita el fallo común de confundirlo con el sistema anfitrión.

Es necesario configurar el *apt* con las fuentes (*sources*) de la rama estable de Debian, que en este momento se trata de Debian Etch. Se edita el fichero */etc/apt/sources.list* y se añaden las fuentes (ver página 246.).

A continuación se actualiza la lista de paquetes:

```
# apt-get update
```

Y después se han de descargar los siguientes paquetes:

- ***localeconf***: configura las variables del sistema para utilizar el conjunto de caracteres seleccionado segun el idioma.
- ***vim***: instala el editor Vim Improved, más cómodo de utilizar.
- ***less***: herramienta similar al comando *more*.
- ***grub***: gestor de arranque necesario para que inicie la máquina.
- ***locales***: conjunto de ficheros y herramientas que permiten la localización e internacionalización de programas que utilizan las librerias de C.
- ***console-data***: conjunto de paquetes que incluye definiciones del mapa de teclado, fuentes para la consola y varias codificaciones del conjunto de caracteres a utilizar en el sistema.
- ***linux-image-2.6.18-4-k7***: imagen precompilada del *kernel* de Linux para procesadores Amd K7.

- ***memtest86***: Utilidad para realizar test de memoria.
- ***mdadm***: Utilidad para la gestión de dispositivos RAID en Linux.

A continuación se debe personalizar el sistema mediante varias acciones:

- **Nombre del *host*:** la maquina debe ser nombrada y configurada para actuar como servidor. Mediante el siguiente comando, se le indica el nombre a utilizar:

  ```
  # hostname boyerito
  ```

- **Interfaces de red:** la IP asignada para *Boyerito* es la **163.117.148.246**, se modifica el fichero */etc/network/interfaces* para que utilice dicha IP (ver página 246.).
- **DNS:** *Boyerito* ha de utilizar a **163.117.131.31** (DNS de la Universidad) temporalmente como servidor DNS. Esto se configura en el fichero */etc/resolv.conf* (ver página 246.).
- ***Hosts*:** se edita el fichero */etc/hosts* indicando cual es la IP de *localhost* (ver página 247.).
- ***Fstab*:** se edita el fichero */etc/fstab* para que utilice */dev/md1* como raíz del sistema, */dev/hda1* y */dev/hdc2* como *swap*, y se especifica el directorio en */mnt* para automatizar el montaje del RAID5 */dev/md0* (ver página 247.).
- **Contraseña:** a continuación se introduce la contraseña de *root*:

  ```
  # passwd
  Enter new UNIX password: ******
  Retype new UNIX password: *****
  passwd: password updated successfully
  ```

- ***Locales*:** para poder definir y utilizar valores de localización distintos del inglés, se utiliza el paquete *locales*, en cuya instalación se han seleccionado las siguientes *locales* a utilizar:

  ```
  es_ES@euro ISO-8859-15
  es_ES ISO-8859-1
  es_ES.UTF-8@euro UTF-8
  es_ES.UTF-8 UTF-8
  ```

Se ha de seleccionar, para utilizar en el sistema como *locale* por defecto, la *'es_ES.UTF-8@euro UTF-8'*

- **Mapa del teclado:** para poder utilizar el teclado en español se utiliza el paquete *console-data*, en el cual se selecciona la siguiente configuración:

  ```
  pc / qwerty / Spanish / Standard / Standard
  ```

- ***Kernel* de Linux:** para instalar la versión precompilada del *kernel* Linux 2.6.18-4 se ejecuta el comando:

  ```
  # apt-get install linux-image-2.6.18-4-k7
  ```

  Después hay que definir en el fichero */etc/modules* los módulos a cargar en el sistema en el arranque (ver página 247.).

Fuera del entorno *chroot*:

Se necesita abandonar el entorno *chroot* para poder crear, desde Knoppix el sector de arranque en */dev/md1*. Para ello se debe teclear:

```
# exit
```

Retornando a la *shell* de Knoppix.

A continuación, hay que indicar al *grub* que instale el sector de arranque

```
# grub-install --root-directory=/mnt/raiz --no-floppy /dev/md1
```

Por último, la máquina ha de iniciar con el *kernel 2.6.18-4-k7*. Se ha de indicar en el fichero */mnt/raiz/boot/grub/menu.lst* el *kernel* a iniciar y los parámetros de arranque (ver página 247.).

8.3.2. Reinicio del sistema y prueba del nuevo *kernel*

El sistema esta listo para ser iniciado, se teclea:

```
# reboot
```

y se retira el cd de Knoppix. La maquina reiniciará por tanto con el nuevo *kernel 2.6.18-4-k7*; se ha de comprobar que arranca de una manera correcta y esperar a que aparezca la pantalla de *login*:

```
Debian GNU/Linux 4.0 boyerito tty1


boyerito login:
```

Se introduce *login* y *password* y se comprueba que los interfaces de red están funcionando:

```
# ifconfig
```

Han de aparecer los interfaces, *eth0* y *lo*, estando la máquina, por tanto, plenamente operativa.

8.3.3. Ficheros de configuración de *Boyerito*

Apt: */etc/apt/sources.list*

```
#Etch
deb ftp://ftp.de.debian.org/debian/ etch main non-free contrib
deb-src ftp://ftp.de.debian.org/debian/ etch  main non-free contrib
deb http://security.debian.org/ etch/updates main contrib non-free
deb http://ftp.rediris.es/debian/ etch main contrib non-free
```

Interfaces de red: */etc/network/interfaces*

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# This entry denotes the loopback (127.0.0.1) interface.
auto lo
iface lo inet loopback

# This entry was created during the Debian installation
# (network, broadcast and gateway are optional)
auto eth0
iface eth0 inet static
        address 163.117.148.246
        netmask 255.255.255.0
        network 163.117.148.0
        broadcast 163.117.148.255
        gateway 163.117.148.2
```

DNS: */etc/resolv.conf*

```
search arcos.inf.uc3m.es
nameserver 163.117.131.31
```

Fstab: */etc/fstab*

```
/dev/md1                /                       ext3    defaults        1 1
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
tmpfs                   /dev/shm                tmpfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
sysfs                   /sys                    sysfs   defaults        0 0
/dev/hda1               swap                    swap    defaults        0 0
/dev/hdc1               swap                    swap    defaults        0 0
/dev/md0                /mnt/md0                ext3    noauto,defaults 0 2
/dev/md1                /mnt/raiz               ext3    noauto,defaults 0 0
```

Hosts: */etc/hosts*

```
127.0.0.1       localhost
163.117.148.246 boyerito
```

Módulos: */etc/modules*

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
8139too
raid0
raid456
```

Grub: */boot/grub/menu.lst*

```
#
# Sample boot menu configuration file
#

# Boot automatically after 30 secs.
timeout 8

# By default, boot the first entry.
default 0

# Fallback to the second entry.
fallback 1

# For booting GNU/Linux
title  GNU/Linux
root (hd0,1)
kernel /boot/vmlinuz-2.6.18-4-k7 root=/dev/md1
initrd /boot/initrd.img-2.6.18-4-k7

# For installing GRUB into the hard disk
title Install GRUB into the hard disk
```

```
root    (hd0,0)
setup   (hd0)

# Change the colors.
title Change the colors
color light-green/brown blink-red/blue
```

8.4. Anexo 4 - Ficheros de configuración

8.4.1. Fichero */etc/ldap/slapd.conf*

```
schemacheck     on
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/inetorgperson.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/samba.schema

modulepath      /usr/lib/ldap
moduleload      back_bdb

pidfile         /var/run/slapd/slapd.pid
argsfile        /var/run/slapd/slapd.args


access to attr=shadowLastChange
                by self write
                by * read

access to attr=sambaLMPassword
                by self write
                by * auth
access to attr=sambaNTPassword
                by self write
                by * auth
access to attr=userPassword
                by self write
                by * auth
access to dn.base=""
                by self write
                by self read
                by * auth
access to *
               by dn="uid=consultas,dc=arcos,dc=inf.uc3m.es" read
               by self read
               by * auth

loglevel        0
```

```
schemacheck     on
idletimeout     30
backend         bdb
database        bdb
checkpoint      1024 5
cachesize       10000

suffix          "dc=arcos,dc=inf.uc3m.es"
rootdn          "cn=admin,dc=arcos,dc=inf.uc3m.es"

rootpw  {SSHA}******************************

directory       /var/lib/ldap

# Indices to maintain
index objectClass           eq
index cn                    pres,sub,eq
index sn                    pres,sub,eq
index uid                   pres,sub,eq
index displayName           pres,sub,eq
index uidNumber             eq
index gidNumber             eq
index memberUID             eq
index sambaSID              eq
index sambaPrimaryGroupSID  eq
index sambaDomainName       eq
index default               sub

allow bind_v2
```

8.4.2. Fichero */etc/smbldap-tools/smbldap.conf*

```
# $Source: /opt/cvs/samba/smbldap-tools/smbldap.conf,v $
# $Id: smbldap.conf,v 1.18 2005/05/27 14:28:47 jtournier Exp $
#
# smbldap-tools.conf : Q & D configuration file for smbldap-tools

#  This code was developped by IDEALX (http://IDEALX.org/) and
#  contributors (their names can be found in the CONTRIBUTORS file).
#
#                 Copyright (C) 2001-2002 IDEALX
#
#  This program is free software; you can redistribute it and/or
#  modify it under the terms of the GNU General Public License
#  as published by the Free Software Foundation; either version 2
#  of the License, or (at your option) any later version.
#
#  This program is distributed in the hope that it will be useful,
```

```
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License
#  along with this program; if not, write to the Free Software
#  Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307,
#  USA.

#  Purpose :
#       . be the configuration file for all smbldap-tools scripts

##############################################################################
#
# General Configuration
#
##############################################################################

# Put your own SID. To obtain this number do: "net getlocalsid".
# If not defined, parameter is taking from "net getlocalsid" return
SID="S-1-5-21-3492619381-3135118558-3133272105"

# Domain name the Samba server is in charged.
# If not defined, parameter is taking from smb.conf configuration file
# Ex: sambaDomain="IDEALX-NT"
sambaDomain="ARCOS"

##############################################################################
#
# LDAP Configuration
#
##############################################################################

# Notes: to use to dual ldap servers backend for Samba, you must patch
# Samba with the dual-head patch from IDEALX. If not using this patch
# just use the same server for slaveLDAP and masterLDAP.
# Those two servers declarations can also be used when you have
# . one master LDAP server where all writing operations must be done
# . one slave LDAP server where all reading operations must be done
#   (typically a replication directory)

# Slave LDAP server
# Ex: slaveLDAP=127.0.0.1
# If not defined, parameter is set to "127.0.0.1"
slaveLDAP="127.0.0.1"

# Slave LDAP port
# If not defined, parameter is set to "389"
slavePort="389"
```

```
# Master LDAP server: needed for write operations
# Ex: masterLDAP=127.0.0.1
# If not defined, parameter is set to "127.0.0.1"
masterLDAP="127.0.0.1"

# Master LDAP port
# If not defined, parameter is set to "389"
masterPort="389"

# Use TLS for LDAP
# If set to 1, this option will use start_tls for connection
# (you should also used the port 389)
# If not defined, parameter is set to "1"
ldapTLS="0"

# How to verify the server's certificate (none, optional or require)
# see "man Net::LDAP" in start_tls section for more details
verify=""

# CA certificate
# see "man Net::LDAP" in start_tls section for more details
cafile=""

# certificate to use to connect to the ldap server
# see "man Net::LDAP" in start_tls section for more details
clientcert=""

# key certificate to use to connect to the ldap server
# see "man Net::LDAP" in start_tls section for more details
clientkey=""

# LDAP Suffix
# Ex: suffix=dc=IDEALX,dc=ORG
suffix="dc=arcos,dc=inf.uc3m.es"

# Where are stored Users
# Ex: usersdn="ou=Users,dc=IDEALX,dc=ORG"
# Warning: if 'suffix' is not set here, you must set the full dn for usersdn
usersdn="ou=People,${suffix}"

# Where are stored Computers
# Ex: computersdn="ou=Computers,dc=IDEALX,dc=ORG"
# Warning: if 'suffix' is not set here, you must set the full dn for computersdn
computersdn="ou=People,${suffix}"

# Where are stored Groups
# Ex: groupsdn="ou=Groups,dc=IDEALX,dc=ORG"
# Warning: if 'suffix' is not set here, you must set the full dn for groupsdn
```

```
groupsdn="ou=Groups,${suffix}"

# Where are stored Idmap entries (used if samba is a domain member server)
# Ex: groupsdn="ou=Idmap,dc=IDEALX,dc=ORG"
# Warning: if 'suffix' is not set here, you must set the full dn for idmapdn
idmapdn="ou=Idmap,${suffix}"

# Where to store next uidNumber and gidNumber available for new users and groups
# If not defined, entries are stored in sambaDomainName object.
# Ex: sambaUnixIdPooldn="sambaDomainName=${sambaDomain},${suffix}"
# Ex: sambaUnixIdPooldn="cn=NextFreeUnixId,${suffix}"
sambaUnixIdPooldn="sambaDomainName=ARCOS,${suffix}"

# Default scope Used
scope="sub"

# Unix password encryption (CRYPT, MD5, SMD5, SSHA, SHA, CLEARTEXT)
hash_encrypt="MD5"

# if hash_encrypt is set to CRYPT, you may set a salt format.
# default is "%s", but many systems will generate MD5 hashed
# passwords if you use "$1$%.8s". This parameter is optional!
crypt_salt_format="%s"

##############################################################################
#
# Unix Accounts Configuration
#
##############################################################################

# Login defs
# Default Login Shell
# Ex: userLoginShell="/bin/bash"
userLoginShell="/bin/bash"

# Home directory
# Ex: userHome="/home/%U"
userHome="/mnt/home/%U"

# Default mode used for user homeDirectory
userHomeDirectoryMode="700"

# Gecos
userGecos="System User"

# Default User (POSIX and Samba) GID
defaultUserGid="513"

# Default Computer (Samba) GID
```

```
defaultComputerGid="515"

# Skel dir
skeletonDir="/mnt/home/skel"

# Default password validation time (time in days) Comment the next line if
# you don't want password to be enable for defaultMaxPasswordAge days (be
# careful to the sambaPwdMustChange attribute's value)
defaultMaxPasswordAge="45"

################################################################################
#
# SAMBA Configuration
#
################################################################################

# The UNC path to home drives location (%U username substitution)
# Just set it to a null string if you want to use the smb.conf 'logon home'
# directive and/or disable roaming profiles
# Ex: userSmbHome="\\PDC-SMB3\%U"
userSmbHome="\\disco\homes"

# The UNC path to profiles locations (%U username substitution)
# Just set it to a null string if you want to use the smb.conf 'logon path'
# directive and/or disable roaming profiles
# Ex: userProfile="\\PDC-SMB3\profiles\%U"
userProfile="\\disco\profiles"

# The default Home Drive Letter mapping
# (will be automatically mapped at logon time if home directory exist)
# Ex: userHomeDrive="H:"
userHomeDrive="Z:"

# The default user netlogon script name (%U username substitution)
# if not used, will be automatically username.cmd
# make sure script file is edited under dos
# Ex: userScript="startup.cmd" # make sure script file is edited under dos
userScript="netlogon.bat"

# Domain appended to the users "mail"-attribute
# when smbldap-useradd -M is used
# Ex: mailDomain="idealx.com"
mailDomain="arcos.inf.uc3m.es"

################################################################################
#
# SMBLDAP-TOOLS Configuration (default are ok for a RedHat)
#
################################################################################
```

```
# Allows not to use smbpasswd (if with_smbpasswd == 0 in smbldap_conf.pm) but
# prefer Crypt::SmbHash library
with_smbpasswd="0"
smbpasswd="/usr/bin/smbpasswd"

# Allows not to use slappasswd (if with_slappasswd == 0 in smbldap_conf.pm)
# but prefer Crypt:: libraries
with_slappasswd="0"
slappasswd="/usr/sbin/slappasswd"

# comment out the following line to get rid of the default banner
# no_banner="1"
```

8.4.3. Fichero */etc/smbldap-tools/smbldap_bind.conf*

```
############################
# Credential Configuration #
############################
# Notes: you can specify two differents configuration if you use a
# master ldap for writing access and a slave ldap server for reading access
# By default, we will use the same DN (so it will work for standard Samba
# release)
slaveDN="cn=admin,dc=arcos,dc=inf.uc3m.es"
slavePw="********"
masterDN="cn=admin,dc=arcos,dc=inf.uc3m.es"
masterPw="*******"
masterDN="cn=admin,dc=arcos,dc=inf.uc3m.es"
masterPw="*******"
```

8.4.4. Fichero */etc/samba/smb.conf* de *Piolin*

```
# Global parameters
[global]
        workgroup = ARCOS
        netbios name = piolin
        enable privileges = yes
        server string = ARCOS Samba PDC Server

        passdb backend = ldapsam:ldap://localhost/
        ldap admin dn = cn=admin,dc=arcos,dc=inf.uc3m.es
        ldap suffix = dc=arcos,dc=inf.uc3m.es
        ldap group suffix = ou=Groups
        ldap user suffix = ou=People
        ldap machine suffix = ou=People
        ldap idmap suffix = ou=Idmap
        ldap ssl = no
```

```
idmap backend = ldap:ldap://localhost
idmap uid = 10000-20000
idmap gid = 10000-20000
#map acl inherit = Yes

add user script = /usr/local/sbin/smbldap-useradd -m "%u"
ldap delete dn = Yes

add machine script = /usr/local/sbin/smbldap-useradd -w "%u"
add group script = /usr/local/sbin/smbldap-groupadd -p "%g"

add user to group script = /usr/local/sbin/smbldap-groupmod -m "%u" "%g"
delete user from group script = /usr/local/sbin/smbldap-groupmod -x "%u" "%g"
set primary group script = /usr/local/sbin/smbldap-usermod -g "%g" "%u"


interfaces = eth0, lo

admin users= @"Domain Admins"
security = user
encrypt passwords = true
min passwd length = 3
obey pam restrictions = No
ldap passwd sync = Yes
mangling method = hash2
log level = 1
syslog = 0
log file = /var/log/samba/log.%m
max log size = 100000

time server = Yes
dos filetimes = yes
fake directory create times = yes
dos filetime resolution = yes

socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
Dos charset = 850
Unix charset = ISO8859-15

logon script = \\disco\netlogon\netlogon.bat
logon drive = H:
logon home = \\disco\homes
logon path = \\disco\profiles

domain logons = Yes
os level = 65
preferred master = Yes
domain master = Yes
local master = Yes
```

```
        dns proxy = no
        wins proxy = no
        wins support = yes
        name resolve order = wins hosts bcast dns



        ; to maintain capital letters in shortcuts in any of the profile folders:
        preserve case = yes
        short preserve case = yes
        case sensitive = no
        use spnego = yes
        hosts allow = 163.117.148.


        client schannel = Auto
        server schannel = Auto
        client signing = auto
        server signing = auto
        winbind trusted domains only = No

[netlogon]
        path = /mnt/home/netlogon/
        browseable = No
        write list = "@Domain Admins"
        guest ok = yes
```

8.4.5. Fichero */etc/samba/smb.conf* de *Lucas*

```
# Global parameters
[global]
        workgroup = ARCOS
        netbios name = disco
        server string = ARCOS Samba Home Directories Server


        passdb backend = ldapsam:ldap://piolin.arcos.inf.uc3m.es/
        ldap admin dn = cn=admin,dc=arcos,dc=inf.uc3m.es
        ldap suffix = dc=arcos,dc=inf.uc3m.es
        ldap group suffix = ou=Groups
        ldap user suffix = ou=People
        ldap machine suffix = ou=People
        ldap idmap suffix = ou=Idmap
        ldap ssl = no

        idmap backend = ldap:ldap://piolin.arcos.inf.uc3m.es
        idmap uid = 10000-20000
        idmap gid = 10000-20000
```

```
        interfaces = eth0, lo
        admin users= @"Domain Admins"
        security = user
        encrypt passwords = true
        obey pam restrictions = No
        ldap passwd sync = Yes
        mangling method = hash2
        log level = 1
        syslog = 0
        log file = /var/log/samba/log.%m
        max log size = 100000

        socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
        Dos charset = 850
        Unix charset = ISO8859-15

        logon script = netlogon.bat
        logon drive = H:
        logon home = \\disco\homes
        logon path = \\disco\profiles

        domain logons = Yes
        os level = 65
        preferred master = Yes
        domain master = no
        dns proxy = no
        wins proxy = no
        wins server = piolin.arcos.inf.uc3m.es
        #name resolve order = wins hosts bcast dns

        preserve case = yes
        short preserve case = yes
        case sensitive = no
        use spnego = yes
        hosts allow = 163.117.148.


[homes]
        comment = Directorio HOME de %U, %u
        read only = No
        create mask = 0640
        directory mask = 0740
        force directory mode = 0770
        browseable = No
        writable = yes
        # next line is a great way to secure the profiles
        force user = %U
        # next line allows administrator to access all profiles
        valid users = %U, "@Domain Admins"
```

```
[netlogon]
        path = /mnt/usuarios/home/netlogon/
        browseable = No
        write list = "@Domain Admins"
        guest ok = yes

#[profiles]
#        read only = no
#        create mask = 0640
#        directory mask = 0700
#        browseable = No
#       profile acls = Yes
#       veto files = desktop.ini
#        guest ok = no
#       csc policy = disable
        # next line is a great way to secure the profiles
#       force user = %U
        # next line allows administrator to access all profiles
#       valid users = %U, "@Domain Admins"

[backup]
        path = /mnt/backup_usuarios/%U
        comment = Directorio de Backup de %U, %u
        read only = No
        create mask = 0640
        directory mask = 0740
        force directory mode = 0770
        browseable = No
        writable = yes
        # next line is a great way to secure the profiles
        force user = %U
        # next line allows administrator to access all profiles
        valid users = %U, "@Domain Admins"
```

8.4.6. Fichero */etc/samba/smb.conf* de *Caponata*

```
# Global parameters
[global]
        workgroup = ARCOS
        netbios name = ssh
        server string = ARCOS Samba SSH Server

        passdb backend = ldapsam:ldap://piolin.arcos.inf.uc3m.es/
        ldap admin dn = cn=admin,dc=arcos,dc=inf.uc3m.es
        ldap suffix = dc=arcos,dc=inf.uc3m.es
        ldap group suffix = ou=Groups
        ldap user suffix = ou=People
        ldap machine suffix = ou=People
```

```
ldap idmap suffix = ou=Idmap
ldap ssl = no

idmap backend = ldap:ldap://piolin.arcos.inf.uc3m.es
idmap uid = 10000-20000
idmap gid = 10000-20000

interfaces = lo
admin users= @"Domain Admins"
security = user
encrypt passwords = true
obey pam restrictions = No
ldap passwd sync = Yes
mangling method = hash2
log level = 1
syslog = 0
log file = /var/log/samba/log.%m
max log size = 100000

socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
Dos charset = 850
Unix charset = ISO8859-15

logon script = netlogon.bat
logon drive = H:
logon home = \\disco\homes
logon path = \\disco\profiles

domain logons = Yes
os level = 65
preferred master = Yes
domain master = no
dns proxy = no
wins proxy = no
wins support = Yes
name resolve order = wins hosts bcast dns

preserve case = yes
short preserve case = yes
case sensitive = no
use spnego = yes
hosts allow = 163.117.148.
```

8.4.7. Fichero */etc/nsswitch.conf*

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the ‘glibc-doc’ and ‘info’ packages installed, try:
```

```
# ‘info libc "Name Service Switch"’ for information about this file.

passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

8.4.8. Fichero */etc/libnss-ldap.conf*

```
host 163.117.148.241

base dc=arcos,dc=inf.uc3m.es

ldap_version 3

binddn uid=consultas,dc=arcos,dc=inf.uc3m.es

bindpw *******

scope one

nss_base_passwd ou=People,dc=arcos,dc=inf.uc3m.es
nss_base_shadow ou=People,dc=arcos,dc=inf.uc3m.es
nss_base_group  ou=Groups,dc=arcos,dc=inf.uc3m.es
nss_base_hosts  ou=People,dc=arcos,dc=inf.uc3m.es
```

8.4.9. Fichero */etc/pam_ldap.conf*

```
host 163.117.148.241

base dc=arcos,dc=inf.uc3m.es

ldap_version 3

binddn uid=consultas,dc=arcos,dc=inf.uc3m.es

bindpw *******
```

```
scope one

nss_base_passwd ou=People,dc=arcos,dc=inf.uc3m.es
nss_base_shadow ou=People,dc=arcos,dc=inf.uc3m.es
nss_base_group  ou=Groups,dc=arcos,dc=inf.uc3m.es
nss_base_hosts  ou=People,dc=arcos,dc=inf.uc3m.es
```

8.4.10. Fichero */etc/pam.d/common-account*

```
#
# /etc/pam.d/common-account - authorization settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authorization modules that define
# the central access policy for use on the system.  The default is to
# only deny service to users whose accounts are expired in /etc/shadow.
#
account sufficient      pam_ldap.so
account required        pam_unix.so
```

8.4.11. Fichero */etc/pam.d/common-auth*

```
#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.).  The default is to use the
# traditional Unix authentication mechanisms.
#
auth    sufficient      pam_ldap.so
auth    required        pam_unix.so try_first_pass
```

8.4.12. Fichero */etc/pam.d/common-password*

```
#
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define  the services to be
#used to change user passwords.  The default is pam_unix

password        sufficient      pam_ldap.so
password        required        pam_unix.so md5 try_first_pass
```

8.4.13. Fichero */etc/pam.d/ssh*

```
#%PAM-1.0
auth       required     pam_nologin.so
#auth       sufficient   pam_ldap.so
auth       required     pam_unix_auth.so try_first_pass
#account    sufficient   pam_ldap.so
account    required     pam_unix_acct.so
#password   sufficient   pam_ldap.so
session    required     pam_unix_session.so
session    optional     pam_mail.so standard noenv
```

8.4.14. Fichero */etc/exports*

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).


#Para piolin (servidor de autenticacion y desde donde se crean los directorios)
/mnt/usuarios/home   163.117.148.241(rw,sync,no_root_squash,subtree_check)
/mnt/usuarios/mail   163.117.148.241(rw,sync,no_root_squash,subtree_check)
/mnt/usuarios/web    163.117.148.241(rw,sync,no_root_squash,subtree_check)
/mnt/backup_usuarios 163.117.148.241(rw,sync,no_root_squash,subtree_check)

#Para piojito ( servidor web y de correo respecto al uso de las cuentas )
/mnt/usuarios/mail 163.117.148.240(rw,sync,root_squash,subtree_check)
/mnt/usuarios/web  163.117.148.240(rw,sync,root_squash,subtree_check)

#Para caponata ( servidor ssh )
/mnt/usuarios/home   163.117.148.245(rw,sync,root_squash,subtree_check)
/mnt/usuarios/mail   163.117.148.245(rw,sync,root_squash,subtree_check)
/mnt/usuarios/web    163.117.148.245(rw,sync,root_squash,subtree_check)
/mnt/backup_usuarios 163.117.148.245(rw,sync,root_squash,subtree_check)

#Para donald ( máquina física desde donde se ejecutan las maquinas virtuales )
/mnt/usuarios        163.117.148.242(ro,sync,no_root_squash,subtree_check)
/mnt/backup_usuarios 163.117.148.242(ro,sync,no_root_squash,subtree_check)
/mnt/sistema         163.117.148.242(ro,sync,no_root_squash,subtree_check)

#Para daisy ( máquina física desde donde se ejecutan las maquinas virtuales )
/mnt/usuarios        163.117.148.243(ro,sync,no_root_squash,subtree_check)
/mnt/backup_usuarios 163.117.148.243(ro,sync,no_root_squash,subtree_check)
/mnt/sistema         163.117.148.243(ro,sync,no_root_squash,subtree_check)

#Para el resto de maquinas
/mnt/usuarios/home/alumnos 163.117.148.0/24(rw,sync,root_squash,subtree_check)
/mnt/usuarios/mail/alumnos 163.117.148.0/24(rw,sync,root_squash,subtree_check)
/mnt/usuarios/web/alumnos  163.117.148.0/24(rw,sync,root_squash,subtree_check)
```

8.4.15. Fichero *listado.sh*

```
#!/bin/sh
# Produce un listado con todas las cuentas de usuario del directorio ldap
# Francisco Olcina Grande 31/10/2006

ldapsearch -D "uid=consultas,dc=arcos,dc=inf.uc3m.es" -x -w XXXXXXX \
-b "ou=People,dc=arcos,dc=inf.uc3m.es" uid|grep "uid:" |awk '{print $2}'
```

8.4.16. Fichero *actualizacion.sh*

```
#!/bin/bash
#Script para actualizar los datos de las cuentas en el LDAP, creación de directorios
y configuración del correo y web de los usuarios.
#set -x

LISTA=$(cat nombres.txt|xargs echo)

echo "Sincronizando con el /var/mail de aguilay"
rsync -au4 --delete aguilaY:/var/mail/* /var/mail/.

for i in $LISTA; do

SHELL=$(smbldap-usershow $i| grep loginShell|awk '{print $2}')
HOME=$(smbldap-usershow $i| grep homeDirectory|awk '{print $2}')
GROUP=$(smbldap-usershow $i |grep gidNumber| awk '{print $2}')
echo "Usuario: $i"
echo "Grupo: $GROUP"
echo "######### Actualizando datos de $i ################################"
smbldap-usermod -C '\\disco\homes' -D 'Z:' -E 'netlogon.bat' -F '\\disco\profiles' $i

if [ "$SHELL" = "/bin/tcsh" ];then
echo "Actualizando shell en LDAP..."
smbldap-usermod -s '/bin/bash' $i
fi

echo "Actualizando home en LDAP..."
HOME_NUEVO=/mnt/home/$(echo $HOME|awk -F '/export/home/aguila/' '{print $2}')
smbldap-usermod -d $HOME_NUEVO $i
MAIL=/mnt/mail/$(echo $HOME|awk -F '/export/home/aguila/' '{print $2}')
WEB=/mnt/web/$(echo $HOME|awk -F '/export/home/aguila/' '{print $2}')

HOME=$HOME_NUEVO

echo "Shell: $SHELL"
echo "Home: $HOME"
echo "Creando directorios de correo y web .."
echo "Directorio de mail: $MAIL"
echo "Directorio de web: $WEB"
```

```
mkdir -p $MAIL/Maildir
chown -R "$i:$GROUP" $MAIL
chmod -R 711 $MAIL

#Creamos los directorios en la cuenta de usuario por si no existen
mkdir -p $HOME/.web
chown -R "$i:$GROUP" $HOME/.web
mkdir -p $HOME/.sweb
chown -R "$i:$GROUP" $HOME/.sweb

#Debe existir el directorio $WEB
mkdir -p $WEB
chown -R "$i:$GROUP" $WEB
chmod 711 $WEB

#No deben existir los directorios en la localización final
rm -fr $WEB/public_html
rm -fr $WEB/private_html

echo "Convirtiendo los correos en formato mbox a maildir.."
echo "1.correos de /var/mail"
su - $i -c "mb2md -m -d $MAIL/Maildir"

echo "2.correos de la carpeta mail (squirrelmail)"
su - $i -c "mb2md -R -s ~/mail -d $MAIL/Maildir"

echo "3.Correos del fichero mbox"
su - $i -c "mb2md -s ~/mbox -d $MAIL/Maildir"

echo "Moviendo las páginas web a la nueva localización"
mv $HOME/.web $WEB/public_html
mv $HOME/.sweb $WEB/private_html

#Enlaces simbólicos necesarios para el apache
su - $i -c "ln -sf $WEB/public_html $HOME/public_html"
su - $i -c "ln -sf $WEB/private_html $HOME/private_html"

#Para piojito, que monta el /mnt/mail como si fuese el home
su - $i -c "ln -sf $WEB/public_html $MAIL/public_html"
su - $i -c "ln -sf $WEB/private_html $MAIL/private_html"

cat > $WEB/private_html/.htaccess <<EOF
AuthName "Zona restringida"
AuthLDAPEnabled on
AuthLDAPAuthoritative on
AuthLDAPBindDN uid=consultas,dc=arcos,dc=inf.uc3m.es
AuthLDAPBindPassword XXXXXXXXXX
AuthLDAPURL ldap://piolin.arcos.inf.uc3m.es:389/ou=People,
    dc=arcos,dc=inf.uc3m.es?uid?sub?(objectClass=*)
```

```
#AuthLDAPGroupAttribute memberuid
#AuthLDAPGroupAttributeIsDN off
AuthType Basic
Require user $i
Options +Indexes
EOF

  chown -R "$USUARIO:$GROUP" $WEB/private_html/.htaccess
  chmod 655 $WEB/private_html/.htaccess

  #Directorio backup
  mkdir $BACKUP
  chown $i $BACKUP
  chmod 700 $BACKUP
  ln -s $BACKUP $HOME/BACKUP
echo "###################### FIN ($i) ################################"

done
```

8.4.17. Fichero *cuentas.sh*

```
#!/bin/bash
#Script para dar de alta una cuenta
#set -x

CORRECTO=1
while [ $CORRECTO -ne 0 ]; do
        NOMBRE=$(dialog --stdout --inputbox 'Introduzca el nombre completo' 0 0)
        USUARIO=$(dialog --stdout --inputbox 'Introduzca el LOGIN que tendrá el usuario' 0 0)
        GROUP=$(dialog --stdout --menu 'Seleccione el grupo del usuario' 0 0 0 \
                1000 'Docencia' \
                2000 'Profesores' \
                3000 'Becarios' \
                4000 'Proyectos' \
                5000 'Alumnos' \
                6000 'Otros' \
                7000 'Masters' \
                8000 'Invitados' )
        RUTA=""
        case $GROUP in
          1000) RUTA=docencia
                ;;
          2000) RUTA=profesores
                ;;
          3000) RUTA=becarios
                ;;
          4000) RUTA=proyectos
                ;;
          5000) RUTA=alumnos
```

```
                ;;
          6000) RUTA=otros
                ;;
          7000) RUTA=masters
                ;;
          8000) RUTA=invitados
                ;;
        esac
        HOME=$(dialog --stdout --inputbox 'Modifique o acepte la ruta del usuario' 0 0
                                                        "/mnt/home/$RUTA/$USUARIO" )

        dialog --yesno "¿Son correctos los datos del usuario?
        Nombre: $NOMBRE
        Login: $USUARIO
        Grupo: $RUTA
        Home: $HOME" 0 0

        CORRECTO=$?
done


smbldap-useradd -a -g $GROUP -d $HOME -s '/bin/bash' -m -C '\\disco\homes' -D 'Z:'
                            -E 'netlogon.bat' -F '\\disco\profiles' -c "$NOMBRE" $USUARIO
echo "Introduzca una contraseña"
smbldap-passwd $USUARIO

MAIL=/mnt/mail/$(echo $HOME|awk -F '/mnt/home/' '{print $2}')
WEB=/mnt/web/$(echo $HOME|awk -F '/mnt/home/' '{print $2}')
BACKUP="/mnt/backup/$USUARIO"

dialog --infobox "Creando directorios de correo y web:
        Directorio de mail: $MAIL
        Directorio de web: $WEB
        Directorio de backup: $BACKUP" 0 0

mkdir -p $MAIL/Maildir
chown -R "$USUARIO:$GROUP" $MAIL
chmod -R 711 $MAIL

mkdir -p $WEB
chown -R "$USUARIO:$GROUP" $WEB
chmod 711 $WEB

mkdir -p $WEB/public_html
mkdir -p $WEB/private_html
chown -R "$USUARIO:www-data" $WEB/public_html
chown -R "$USUARIO:www-data" $WEB/private_html
chmod 750 $WEB/public_html
chmod 750 $WEB/private_html
```

```
su - $USUARIO -c "ln -sf $WEB/public_html $HOME/public_html"
su - $USUARIO -c "ln -sf $WEB/private_html $HOME/private_html"

#Para piojito, que monta el /mnt/mail como si fuese el home

su - $USUARIO -c "ln -sf $WEB/public_html $MAIL/public_html"
su - $USUARIO -c "ln -sf $WEB/private_html $MAIL/private_html"

cat > $WEB/private_html/.htaccess <<EOF
AuthName "Zona restringida"
AuthLDAPEnabled on
AuthLDAPAuthoritative on
AuthLDAPBindDN uid=consultas,dc=arcos,dc=inf.uc3m.es
AuthLDAPBindPassword XXXXXX
AuthLDAPURL ldap://piolin.arcos.inf.uc3m.es:389/ou=People,dc=arcos,dc=inf.uc3m.es?
                                                        uid?sub?(objectClass=*)
#AuthLDAPGroupAttribute memberuid
#AuthLDAPGroupAttributeIsDN off
AuthType Basic
Require user $USUARIO
Options +Indexes
EOF

chown -R "$USUARIO:$GROUP" $WEB/private_html/.htaccess
chmod 655 $WEB/private_html/.htaccess

#Directorio backup
mkdir $BACKUP
chown $USUARIO $BACKUP
chmod 700 $BACKUP
ln -s $BACKUP $HOME/BACKUP

mail -s "Bienvenido a ARCOS" $USUARIO@arcos.inf.uc3m.es < /dev/null
```

8.4.18. Fichero */etc/apache/httpd.conf*

```
ServerType standalone

ServerRoot /etc/apache

LockFile /var/lock/apache.lock

PidFile /var/run/apache.pid

ScoreBoardFile /var/run/apache.scoreboard

Timeout 300
```

```
KeepAlive On

MaxKeepAliveRequests 100

KeepAliveTimeout 15

MinSpareServers 5
MaxSpareServers 10

StartServers 5

MaxClients 150

MaxRequestsPerChild 100

Include /etc/apache/modules.conf

<IfModule mod_status.c>
  ExtendedStatus On
</IfModule>

Port 80

User www-data
Group www-data

ServerAdmin webmaster@localhost

ServerName localhost

DocumentRoot /mnt/home/otros/web_en/public_html/

<Directory />
    Options FollowSymLinks -SymLinksIfOwnerMatch
    AllowOverride all
</Directory>

<Directory /mnt/home/otros/web_en/public_html/>

    Options -Indexes Includes FollowSymLinks MultiViews

    AllowOverride options

    Order allow,deny
    Allow from all
</Directory>

<IfModule mod_userdir.c>
    UserDir public_html
```

```
</IfModule>

<Directory /mnt/home/*/*/public_html>
    AllowOverride FileInfo AuthConfig Limit Options
    Options MultiViews SymLinksIfOwnerMatch +Includes -Indexes
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>
<Directory /mnt/web/*/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews SymLinksIfOwnerMatch +Includes
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>
<Directory /mnt/web/*/*/public_html/cgi-bin>
        AllowOverride None
        order allow,deny
        Deny from all
</Directory>
<Directory /mnt/home/*/*/public_html/cgi-bin>
        AllowOverride None
        order allow,deny
        Deny from all
</Directory>

<IfModule mod_perl.c>
    <Directory /mnt/web/*/*/public_html/perl/>
        Deny from all
    </Directory>
</IfModule>

<Directory /mnt/home/*/*/*/public_html>
    AllowOverride FileInfo AuthConfig Limit Options
    Options MultiViews SymLinksIfOwnerMatch +Includes -Indexes
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
```

```
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>
<Directory /mnt/web/*/*/*/public_html>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews SymLinksIfOwnerMatch +Includes
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>
<Directory /mnt/web/*/*/*/public_html/cgi-bin>
        AllowOverride None
        order allow,deny
        Deny from all
</Directory>
<Directory /mnt/home/*/*/*/public_html/cgi-bin>
        AllowOverride None
        order allow,deny
        Deny from all
</Directory>

<IfModule mod_perl.c>
    <Directory /mnt/web/*/*/*/public_html/perl/>
        Deny from all
    </Directory>
</IfModule>

<Location /index.shtml>
    Options +Includes
</Location>

<IfModule mod_dir.c>
    DirectoryIndex index.html index.htm index.shtml index.cgi index.php
</IfModule>

AccessFileName .htaccess

<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
```

```
UseCanonicalName Off

TypesConfig /etc/mime.types

DefaultType text/plain

<IfModule mod_mime_magic.c>
    MIMEMagicFile /usr/share/misc/file/magic.mime
</IfModule>

HostnameLookups Off

ErrorLog /var/log/apache/error.log

LogLevel warn

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"
                                          \"%{forensic-id}n\" %T %v" full
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"
                                        \"%{forensic-id}n\" %P %T" debug
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"
                                          \"%{forensic-id}n\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{forensic-id}n\"" forensic
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

CustomLog /var/log/apache/access.log combined

<IfModule mod_log_forensic.c>
 ForensicLog /var/log/apache/forensic.log
</IfModule>

<IfModule mod_backtrace.c>
 EnableExceptionHook On
</IfModule>

<IfModule mod_whatkilledus.c>
 EnableExceptionHook On
</IfModule>

ServerSignature On

<IfModule mod_alias.c>
    Alias /icons/ /usr/share/apache/icons/

    <Directory /usr/share/apache/icons>
         Options Indexes MultiViews
```

```
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>

    Alias /images/ /usr/share/images/

    <Directory /usr/share/images>
        Options MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
</IfModule>

<IfModule mod_alias.c>
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/

    <Directory /usr/lib/cgi-bin/>
        AllowOverride None
        Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
    <Directory /usr/lib/cgi-bin/admin>
        AllowOverride all
        Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
</IfModule>

<IfModule mod_autoindex.c>

    IndexOptions FancyIndexing NameWidth=*

    AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

    AddIconByType (TXT,/icons/text.gif) text/*
    AddIconByType (IMG,/icons/image2.gif) image/*
    AddIconByType (SND,/icons/sound2.gif) audio/*
    AddIconByType (VID,/icons/movie.gif) video/*

    AddIcon /icons/binary.gif .bin .exe
    AddIcon /icons/binhex.gif .hqx
    AddIcon /icons/tar.gif .tar
    AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
    AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
    AddIcon /icons/a.gif .ps .ai .eps
```

```
    AddIcon /icons/layout.gif .html .shtml .htm .pdf
    AddIcon /icons/text.gif .txt
    AddIcon /icons/c.gif .c
    AddIcon /icons/p.gif .pl .py
    AddIcon /icons/f.gif .for
    AddIcon /icons/dvi.gif .dvi
    AddIcon /icons/uuencoded.gif .uu
    AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
    AddIcon /icons/tex.gif .tex
    AddIcon /icons/bomb.gif core
    AddIcon /icons/deb.gif .deb

    AddIcon /icons/back.gif ..
    AddIcon /icons/hand.right.gif README
    AddIcon /icons/folder.gif ^^DIRECTORY^^
    AddIcon /icons/blank.gif ^^BLANKICON^^

    DefaultIcon /icons/unknown.gif

    ReadmeName README.html
    HeaderName HEADER.html

</IfModule>

<IfModule mod_mime.c>

    AddEncoding x-compress Z
    AddEncoding x-gzip gz tgz

    AddLanguage da .dk
    AddLanguage nl .nl
    AddLanguage en .en
    AddLanguage et .ee
    AddLanguage fr .fr
    AddLanguage de .de
    AddLanguage el .el
    AddLanguage it .it
    AddLanguage ja .ja
    AddCharset ISO-2022-JP .jis
    AddLanguage pl .po
    AddCharset ISO-8859-2 .iso-pl
    AddLanguage pt .pt
    AddLanguage pt-br .pt-br
    AddLanguage lb .lu
    AddLanguage ca .ca
    AddLanguage es .es
    AddLanguage sv .se
    AddLanguage cs .cz
```

```
    <IfModule mod_negotiation.c>
        LanguagePriority en da nl et fr de el it ja pl pt pt-br lb ca es sv
    </IfModule>

    AddType application/x-tar .tgz
    AddType image/bmp .bmp

    AddType text/x-hdml .hdml

    <IfModule mod_include.c>
        AddType text/html .shtml
        AddHandler server-parsed .shtml
    </IfModule>

</IfModule>

AddDefaultCharset on

ErrorDocument 404 /errores/404.php

<IfModule mod_setenvif.c>
    BrowserMatch "Mozilla/2" nokeepalive
    BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0

    BrowserMatch "RealPlayer 4\.0" force-response-1.0
    BrowserMatch "Java/1\.0" force-response-1.0
    BrowserMatch "JDK/1\.0" force-response-1.0
</IfModule>

<IfModule mod_perl.c>
  <IfModule mod_alias.c>
   Alias /perl/ /mnt/home/otros/web/public_html/perl/
  </IfModule>
  <Location /perl>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
  </Location>
</IfModule>

<IfModule mod_alias.c>
 Alias /doc/ /usr/share/doc/
</IfModule>

<Location /doc>
  order deny,allow
  deny from all
  allow from 127.0.0.0/255.0.0.0
  Options Indexes FollowSymLinks MultiViews
```

```
</Location>

<IfModule mod_proxy.c>

</IfModule>

NameVirtualHost *

<VirtualHost *>
ServerName     arcos.inf.uc3m.es
DocumentRoot   /mnt/home/otros/web_en/public_html/
ErrorLog       /var/log/apache/arcos.inf.uc3m.es-error_log
TransferLog    /var/log/apache/arcos.inf.uc3m.es-access_log
</VirtualHost>

<VirtualHost *>
ServerAdmin    acaldero@arcos.inf.uc3m.es
ServerName     www-es.arcos.inf.uc3m.es
DocumentRoot   /mnt/home/otros/web/public_html/
ErrorLog       /var/log/apache/arcos.inf.uc3m.es-error_log
TransferLog    /var/log/apache/arcos.inf.uc3m.es-access_log
</VirtualHost>

<VirtualHost *>
ServerAdmin    acaldero@arcos.inf.uc3m.es
ServerName     www-en.arcos.inf.uc3m.es
DocumentRoot   /mnt/home/otros/web_en/public_html/
ErrorLog       /var/log/apache/arcos.inf.uc3m.es-error_log
TransferLog    /var/log/apache/arcos.inf.uc3m.es-access_log
</VirtualHost>

<VirtualHost *>
ServerAdmin    folcina@arcos.inf.uc3m.es
ServerName     pruebas.arcos.inf.uc3m.es
DocumentRoot   /var/www-pruebas/
ErrorLog       /var/log/apache/pruebas.arcos.inf.uc3m.es-error_log
TransferLog    /var/log/apache/pruebas.arcos.inf.uc3m.es-access_log
</VirtualHost>

<VirtualHost *>
ServerAdmin   jcarrete@arcos.inf.uc3m.es
ServerName    jornadas.arcos.inf.uc3m.es
DocumentRoot  /mnt/web/proyectos/jornadas/public_html
ErrorLog      /var/log/apache/jornadas.arcos.inf.uc3m.es-error_log
TransferLog   /var/log/apache/jornadas.arcos.inf.uc3m.es-access_log
</VirtualHost>

<VirtualHost *>
ServerAdmin   fgarcia@arcos.inf.uc3m.es
```

```
ServerName    winpfs.arcos.inf.uc3m.es
DocumentRoot  /mnt/web/proyectos/winpfs/public_html
ErrorLog      /var/log/apache/winpfs.arcos.inf.uc3m.es-error_log
TransferLog   /var/log/apache/winpfs.arcos.inf.uc3m.es-access_log
</VirtualHost>

<VirtualHost *>
ServerAdmin   fgarcia@arcos.inf.uc3m.es
ServerName    xpn.arcos.inf.uc3m.es
DocumentRoot  /mnt/web/proyectos/xpn/public_html
ErrorLog      /var/log/apache/xpn.arcos.inf.uc3m.es-error_log
TransferLog   /var/log/apache/xpn.arcos.inf.uc3m.es-access_log
</VirtualHost>

<VirtualHost *>
ServerAdmin   acaldero@arcos.inf.uc3m.es
ServerName    mimpi.arcos.inf.uc3m.es
DocumentRoot  /mnt/web/proyectos/mimpi/public_html
ErrorLog      /var/log/apache/mimpi.arcos.inf.uc3m.es-error_log
TransferLog   /var/log/apache/mimpi.arcos.inf.uc3m.es-access_log
</VirtualHost>

<VirtualHost *>
ServerAdmin   jdaniel@arcos.inf.uc3m.es
ServerName    magsi.arcos.inf.uc3m.es
DocumentRoot  /mnt/web/masters/magsi/public_html
ErrorLog      /var/log/apache/magsi.arcos.inf.uc3m.es-error_log
TransferLog   /var/log/apache/magsi.arcos.inf.uc3m.es-access_log
</VirtualHost>

<VirtualHost *>
ServerAdmin   acaldero@arcos.inf.uc3m.es
ServerName    cupcam.arcos.inf.uc3m.es
DocumentRoot  /mnt/home/otros/cupcam/public_html
ErrorLog      /var/log/apache/cupcam.arcos.inf.uc3m.es-error_log
TransferLog   /var/log/apache/cupcam.arcos.inf.uc3m.es-access_log
</VirtualHost>

Include /etc/apache/conf.d
```

8.4.19. Fichero */etc/apache-ssl/httpd.conf*

```
ServerType standalone

ServerRoot /etc/apache-ssl

LockFile /var/lock/apache-ssl.lock

PidFile /var/run/apache-ssl.pid
```

```
ScoreBoardFile /var/run/apache-ssl.scoreboard

Timeout 300

KeepAlive On

MaxKeepAliveRequests 100

KeepAliveTimeout 15

MinSpareServers 5
MaxSpareServers 10

StartServers 5

MaxClients 150

MaxRequestsPerChild 100

Listen 443

Include /etc/apache-ssl/modules.conf

<IfModule mod_status.c>
  ExtendedStatus On
</IfModule>

Port 443

User www-data
Group www-data

ServerAdmin webmaster@localhost

ServerName localhost

DocumentRoot /mnt/home/otros/web/private_html/

<Directory />

    Options FollowSymLinks
    AllowOverride All
</Directory>

<IfModule mod_userdir.c>
    UserDir private_html
</IfModule>
```

```
<Directory /mnt/home/*/*/private_html>
    AllowOverride All
    Options MultiViews SymLinksIfOwnerMatch +Includes -Indexes

    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>
<Directory /mnt/web/*/*/private_html>
    AllowOverride All
    Options MultiViews SymLinksIfOwnerMatch +Includes -Indexes

    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>

<Directory /mnt/home/*/*/*/private_html>
    AllowOverride All
    Options MultiViews SymLinksIfOwnerMatch +Includes -Indexes

    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>
<Directory /mnt/web/*/*/*/private_html>
    AllowOverride All
    Options MultiViews SymLinksIfOwnerMatch +Includes -Indexes

    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
```

```
        Order deny,allow
        Deny from all
    </Limit>
</Directory>

<Directory /mnt/web/*/*/private_html/cgi-bin/>
        AllowOverride None
        Order allow,deny
        Deny from all
</Directory>

<Directory /mnt/home/*/*/private_html/cgi-bin/>
        AllowOverride None
        Order allow,deny
        Deny from all
</Directory>

<Directory /mnt/web/*/*/*/private_html/cgi-bin/>
        AllowOverride None
        Order allow,deny
        Deny from all
</Directory>

<Directory /mnt/home/*/*/*/private_html/cgi-bin/>
        AllowOverride None
        Order allow,deny
        Deny from all
</Directory>

<Location /index.shtml>
    Options +Includes
</Location>

<Directory /mnt/home/otros/web/private_html/>

    Options -Indexes Includes FollowSymLinks MultiViews

    AllowOverride all

    Order allow,deny
    Allow from all
</Directory>

<IfModule mod_dir.c>
    DirectoryIndex index.html index.htm index.shtml index.cgi index.php
</IfModule>

AccessFileName .htaccess
```

```
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>

UseCanonicalName Off

TypesConfig /etc/mime.types

DefaultType text/plain

<IfModule mod_mime_magic.c>
    MIMEMagicFile /usr/share/misc/file/magic.mime
</IfModule>

HostnameLookups Off

ErrorLog /var/log/apache-ssl/error.log

LogLevel warn

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" \"%{forensic-id}n\"
                                                                          %T %v" full
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" \"%{forensic-id}n\"
                                                                        %P %T" debug
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" \"%{forensic-id}n\""
                                                                             combined
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{forensic-id}n\"" forensic
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

CustomLog /var/log/apache-ssl/access.log combined

<IfModule mod_log_forensic.c>
 ForensicLog /var/log/apache-ssl/forensic.log
</IfModule>

<IfModule mod_backtrace.c>
 EnableExceptionHook On
</IfModule>

<IfModule mod_whatkilledus.c>
 EnableExceptionHook On
</IfModule>

ServerSignature On

<IfModule mod_alias.c>
```

```
    Alias /icons/ /usr/share/apache/icons/

    <Directory /usr/share/apache/icons>
         Options Indexes MultiViews
         AllowOverride None
         Order allow,deny
         Allow from all
    </Directory>

    Alias /images/ /usr/share/images/

    <Directory /usr/share/images>
         Options MultiViews
         AllowOverride None
         Order allow,deny
         Allow from all
    </Directory>

Alias /Pipermail/ /usr/lib/mailman/archives/public/

</IfModule>

<Directory /usr/lib/mailman/archives/public/>
    Options MultiViews
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>

<IfModule mod_alias.c>
  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  ScriptAlias /mailman/ /usr/lib/cgi-bin/mailman/

    <Directory /usr/lib/cgi-bin/>
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
    <Directory /usr/lib/cgi-bin/admin>
        AllowOverride all
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
</IfModule>

<IfModule mod_autoindex.c>
```

```
    IndexOptions FancyIndexing NameWidth=*

    AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

    AddIconByType (TXT,/icons/text.gif) text/*
    AddIconByType (IMG,/icons/image2.gif) image/*
    AddIconByType (SND,/icons/sound2.gif) audio/*
    AddIconByType (VID,/icons/movie.gif) video/*

    AddIcon /icons/binary.gif .bin .exe
    AddIcon /icons/binhex.gif .hqx
    AddIcon /icons/tar.gif .tar
    AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
    AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
    AddIcon /icons/a.gif .ps .ai .eps
    AddIcon /icons/layout.gif .html .shtml .htm .pdf
    AddIcon /icons/text.gif .txt
    AddIcon /icons/c.gif .c
    AddIcon /icons/p.gif .pl .py
    AddIcon /icons/f.gif .for
    AddIcon /icons/dvi.gif .dvi
    AddIcon /icons/uuencoded.gif .uu
    AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
    AddIcon /icons/tex.gif .tex
    AddIcon /icons/bomb.gif core
    AddIcon /icons/deb.gif .deb

    AddIcon /icons/back.gif ..
    AddIcon /icons/hand.right.gif README
    AddIcon /icons/folder.gif ^^DIRECTORY^^
    AddIcon /icons/blank.gif ^^BLANKICON^^

    DefaultIcon /icons/unknown.gif

    ReadmeName README.html
    HeaderName HEADER.html

</IfModule>

<IfModule mod_mime.c>

    AddEncoding x-compress Z
    AddEncoding x-gzip gz tgz

    AddLanguage da .dk
    AddLanguage nl .nl
    AddLanguage en .en
    AddLanguage et .ee
    AddLanguage fr .fr
```

```
    AddLanguage de .de
    AddLanguage el .el
    AddLanguage it .it
    AddLanguage ja .ja
    AddCharset ISO-2022-JP .jis
    AddLanguage pl .po
    AddCharset ISO-8859-2 .iso-pl
    AddLanguage pt .pt
    AddLanguage pt-br .pt-br
    AddLanguage lb .lu
    AddLanguage ca .ca
    AddLanguage es .es
    AddLanguage sv .se
    AddLanguage cs .cz

    <IfModule mod_negotiation.c>
        LanguagePriority en da nl et fr de el it ja pl pt pt-br lb ca es sv
    </IfModule>

    AddType application/x-httpd-php .php

    AddType application/x-tar .tgz
    AddType image/bmp .bmp

    AddType text/x-hdml .hdml

    <IfModule mod_include.c>
     AddType text/html .shtml
     AddHandler server-parsed .shtml
    </IfModule>

</IfModule>

AddDefaultCharset on

ErrorDocument 404 /errores/404.php

<IfModule mod_setenvif.c>
    BrowserMatch "Mozilla/2" nokeepalive
    BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0

    BrowserMatch "RealPlayer 4\.0" force-response-1.0
    BrowserMatch "Java/1\.0" force-response-1.0
    BrowserMatch "JDK/1\.0" force-response-1.0
</IfModule>

<IfModule mod_perl.c>
  <IfModule mod_alias.c>
   Alias /perl/ /var/www/perl/
```

```
  </IfModule>
  <Location /perl>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
  </Location>
</IfModule>

<IfModule mod_alias.c>
 Alias /doc/ /usr/share/doc/
</IfModule>

<Location /doc>
  order deny,allow
  deny from all
  allow from 127.0.0.0/255.0.0.0
  Options Indexes FollowSymLinks MultiViews
</Location>

<IfModule mod_proxy.c>

</IfModule>

SSLRandomFile file /dev/urandom 1024

SSLRandomFilePerConnection file /dev/urandom 1024

SSLEnable

SSLCacheServerPath /usr/lib/apache-ssl/gcache

SSLCacheServerPort /var/run/gcache_port

SSLSessionCacheTimeout 15

SSLCertificateFile /etc/apache-ssl/apache.pem

SSLVerifyClient 0
SSLVerifyDepth 10

SSLUseCRL

SSLCRLCheckAll

SSLOnRevocationSetEnv SSL_REVOKED

SSLOnCRLExpirySetEnv SSL_CRL_EXPIRED

SSLOnNoCRLSetEnv SSL_NO_CRL
```

```
SSLFakeBasicAuth

CustomLog /var/log/apache-ssl/ssl.log "%t %{version}c %{cipher}c %{clientcert}c"

Include /etc/apache-ssl/conf.d
Include /etc/amavis-stats/apache.conf
```

8.4.20. Fichero */etc/postfix/main.cf* de *Piojito*

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

myhostname = piojito.arcos.inf.uc3m.es
alias_maps = hash:/etc/aliases,hash:/var/lib/mailman/data/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = piojito.arcos.inf.uc3m.es, localhost.arcos.inf.uc3m.es, , localhost,
    arcos.inf.uc3m.es,www.arcos.inf.uc3m.es,www2.arcos.inf.uc3m.es,arcos.inf.uc3m.es
relayhost = smtp.uc3m.es
mynetworks = 127.0.0.0/8 163.117.148.0/24
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
#inet_interfaces = localhost,163.117.148.104
home_mailbox = Maildir/
unknow_local_recipient_reject_code = 550
mailbox_size_limit = 0
 ##10mb de limite
 #message_size_limit = 10485760
#45mb de limite
message_size_limit = 47185920


#####SPAM#########

#Filtro de amavis
content_filter = smtp-amavis:[localhost]:10024

mtpd_client_restrictions = permit_mynetworks,
```

```
        reject_rbl_client sbl.spamhaus.org,

        reject_rbl_client relays.ordb.org,

        reject_rbl_client opm.blitzed.org,

        reject_unauth_destination

smtpd_error_sleep_time = 1s
smtpd_soft_error_limit = 60
smtpd_hard_error_limit = 10
default_process_limit = 3
disable_vrfy_command = yes
smtpd_helo_required = yes

smtpd_recipient_restrictions =
        reject_invalid_hostname,
        #reject_non_fqdn_hostname,
        reject_non_fqdn_sender,
        reject_non_fqdn_recipient,
        reject_unknown_sender_domain,
        reject_unknown_recipient_domain,
        permit_mynetworks,
        permit_sasl_authenticated,
        reject_unauth_destination,
        check_recipient_access pcre:/etc/postfix/recipient_checks.pcre,
        check_helo_access hash:/etc/postfix/helo_checks,
        # check_helo_access pcre:/etc/postfix/helo_checks.pcre,
        check_sender_access hash:/etc/postfix/sender_checks,
        #check_client_access hash:/etc/postfix/client_checks,
        #check_client_access pcre:/etc/postfix/client_checks.pcre,
        reject_rbl_client relays.ordb.org,
        # reject_rbl_client opm.blitzed.org,
        # reject_rbl_client list.dsbl.org,
        # reject_rbl_client sbl.spamhaus.org,
        # reject_rbl_client cbl.abuseat.org,
        # reject_rbl_client dul.dnsbl.sorbs.net,
        permit

smtpd_data_restrictions =
        reject_unauth_pipelining,
        permit
```

8.4.21. Fichero */etc/postfix/main.cf* del resto de máquinas

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
```

```
# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

myhostname = [nombre de la máquina]
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, $myhostname.arcos.inf.uc3m.es,
                     localhost.arcos.inf.uc3m.es, localhost
relayhost = piojito.arcos.inf.uc3m.es
mynetworks = 127.0.0.0/8
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
```

8.4.22. Fichero */etc/bind/named.conf*

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

acl "slaves" {
        163.117.131.31;
        163.117.131.43;
};

include "/etc/bind/named.conf.options";

// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};
```

```
zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

zone "arcos.inf.uc3m.es" {
        type master;
        file "/etc/bind/db.arcos.inf.uc3m.es";
        allow-query {any;};
        allow-transfer {slaves;};
};

zone "148.117.163.in-addr.arpa" {
        type master;
        file "/etc/bind/db.148.117.163";
        allow-query { any; };
        allow-transfer { slaves; };
};


include "/etc/bind/named.conf.local";
```

8.4.23. Fichero */etc/bind/db.arcos.inf.uc3m.es*

```
;
;
; IMPORTANTE:
;  - direcciones de nombres canónicos
;  - las primeras 60 direcciones son del centro de calculo el resto son nuestras
;  - número de serie: <año:4><mes:2><día:2><versión:2>
;    si se cambia el fichero en el mismo día, entonces se incrementa el
;    número de versión en una unidad
;

arcos.inf.uc3m.es. IN SOA piojito.arcos.inf.uc3m.es. root.piojito.arcos.inf.uc3m.es. (
        2007072701      ; Número de Serie
        28800           ; Refresco después de 8 horas
        7200            ; Reintento después de 2 horas
```

```
        604800          ; Expiración después de una semana
        86400 )         ; TTL mínimo de 1 dia


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Configuración de los Servidores de nombres (The name '@' is implied)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                        IN NS piojito.arcos.inf.uc3m.es.
                        IN NS vortex.uc3m.es.


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Configuración de las máquinas del correo
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

arcos.inf.uc3m.es.      IN TXT          "v=spf1 mx ~all"
                        IN MX           5 smtp.uc3m.es.
                        IN MX           5 smtp01.uc3m.es.
                        IN MX           5 smtp02.uc3m.es.
                        IN MX           5 smtp03.uc3m.es.
                        IN MX           8 mail.rediris.es.
arcos.inf.uc3m.es.      IN A            163.117.148.240


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; NOMBRES DEL DOMINIO ARCOS.INF.UC3M.ES
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; ARCOS
aguila                  IN CNAME        aguilaY

;
; Nuevos servidores (Paco y Alex)
;
piojito         IN A    163.117.148.240
piolin          IN A    163.117.148.241
donald          IN A    163.117.148.242
daisy           IN A    163.117.148.243
lucas           IN A    163.117.148.244
caponata        IN A    163.117.148.245
boyerito        IN A    163.117.148.246

(resto de máquinas)
......
```

8.4.24. Fichero */etc/bind/db.148.117.163*

```
148.117.163.in-addr.arpa. IN SOA piojito.arcos.inf.uc3m.es. root.piojito.arcos.inf.uc3m.es. (
```

```
        2007062501      ; Número de Serie
        28800           ; Refresco despu'es de 8 horas
        7200            ; Reintento despu'es de 2 horas
        604800          ; Expiraci'on despu'es de una semana
        86400 )         ; TTL m'inimo de 1 dia

;
; Servidores de nombres (The name '@' is implied)

                          IN NS piojito.arcos.inf.uc3m.es.
                          IN NS vortex.uc3m.es.


; direcciones de nombres canonicos

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; DIRECCIONES DE ARCOS.INF.UC3M.ES
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
240     IN PTR  piojito.arcos.inf.uc3m.es.
241     IN PTR  piolin.arcos.inf.uc3m.es.
242     IN PTR  donald.arcos.inf.uc3m.es.
243     IN PTR  daisy.arcos.inf.uc3m.es.
244     IN PTR  lucas.arcos.inf.uc3m.es.
245     IN PTR  caponata.arcos.inf.uc3m.es.
246     IN PTR  boyerito.arcos.inf.uc3m.es.
```

8.4.25. Fichero */etc/resolv.conf*

```
search arcos.inf.uc3m.es
nameserver 163.117.148.240
nameserver 163.117.131.31
```

8.4.26. Fichero */etc/hosts*

```
127.0.0.1               localhost.localdomain localhost
163.117.148.240         piojito.arcos.inf.uc3m.es piojito
163.117.148.241         piolin.arcos.inf.uc3m.es piolin
163.117.148.242         donald.arcos.inf.uc3m.es donald
163.117.148.243         daisy.arcos.inf.uc3m.es daisy
163.117.148.244         lucas.arcos.inf.uc3m.es lucas
163.117.148.245         caponata.arcos.inf.uc3m.es caponata
163.117.148.246         boyerito.arcos.inf.uc3m.es boyerito
```

8.4.27. Fichero */etc/security/limits.conf*

```
# Evitar que nadie tenga acceso a core files
*       hard    core            0
```

```
# Maximo numero de procesos. Soft limit 15. Hard limit 20
*        soft    nproc          15
*        hard    nproc          20
svn        soft    nproc        50
svn        hard    nproc        70

# Tiempo maximo de proceso que puede lanzar un usuario (en segundos)
# soft = 240min (4 horas) ; hard = 300min (5 horas)
*       soft    cpu             120
*       hard    cpu             150
svn     soft    cpu             -
svn     hard    cpu             -

# Prioridad con la que se lanzan los procesos de los usuarios
# Prioridad en linux varia entre (19 min y -20 max)
# No utilizar 19, ni tampoco ninguna por debajo de -5
# man nice para mas info
*       hard    priority        10
```

8.4.28. Fichero *.htaccess* de la *intranet* de ARCOS

```
AuthName "Zona restringida, introducir nombre de usuario y password de ARCOS"
AuthLDAPEnabled on
AuthLDAPAuthoritative on
AuthLDAPBindDN uid=consultas,dc=arcos,dc=inf.uc3m.es
AuthLDAPBindPassword XXXXXXX
AuthLDAPURL ldap://piolin.arcos.inf.uc3m.es:389/ou=People,
  dc=arcos,dc=inf.uc3m.es?uid?sub?(objectClass=*)
AuthType Basic
Require valid-user
Options +Indexes
```

8.4.29. Fichero *conectividad.sh*

```
#!/bin/sh
#set -x
DESTINO=$1

echo -n "Comprobando conectividad con $DESTINO ... "
/usr/bin/ssh $DESTINO hostname &> /dev/null
RESULTADO=$?

if [ $RESULTADO -eq 0 ]; then
        echo "OK"
else
        echo "FALLO"
fi

exit $RESULTADO
```

8.4.30. Fichero *backup_raiz.sh*

```
#!/bin/sh
# Script que realiza un backup incremental de la máquina pasada como primer argumento

#set -x
MAQUINA_ORIGEN=$1

if [ $MAQUINA_ORIGEN ]; then
        DIR_ORIGEN=/mnt/raiz

        DIR_DESTINO=/mnt/sistema/maquinas/$MAQUINA_ORIGEN
        HORA_COMIENZO=$(date +%d_%B_%H-%M)

        FICH_STDOUT="/root/temp/$MAQUINA_ORIGEN-$HORA_COMIENZO-stdout.txt"
        FICH_STDERR="/root/temp/$MAQUINA_ORIGEN-$HORA_COMIENZO-stderr.txt"
        FICH_LOG_SALIDA=/root/backup/salidas/maquina_$MAQUINA_ORIGEN/$HORA_COMIENZO

        /root/scripts/conectividad.sh $MAQUINA_ORIGEN
        CONECTIVIDAD=$?
        OPCIONES_RSYNC="-aud4z --exclude-from=/root/backup/exclude_raiz --stats --delete
                        --backup --backup-dir=$DIR_DESTINO/$HORA_COMIENZO"

        touch $FICH_STDOUT
        touch $FICH_STDERR

        if [ $CONECTIVIDAD -eq 0 ]; then

                ssh $MAQUINA_ORIGEN "mount $DIR_ORIGEN" 1>> $FICH_STDOUT 2>> $FICH_STDERR
                nice -n 15 rsync $OPCIONES_RSYNC $MAQUINA_ORIGEN:$DIR_ORIGEN
                                $DIR_DESTINO 1>> $FICH_STDOUT 2>> $FICH_STDERR

                ssh $MAQUINA_ORIGEN "umount $DIR_ORIGEN" 1>> $FICH_STDOUT 2>> $FICH_STDERR
        fi


        HORA_FIN=$(date +%d_%B_%H-%M)
        #Generando fichero de salida

        cat > $FICH_LOG_SALIDA <<EOF
HORA DE COMIENZO=$HORA_COMIENZO
HORA FIN=$HORA_FIN
CONEXION CON $MAQUINA_ORIGEN=$(if [ $CONECTIVIDAD -eq 0 ]; then echo "OK"; else echo "FALLO"; fi)
SALIDA ESTANDAR
$(cat $FICH_STDOUT)

SALIDA ERROR
$(cat $FICH_STDERR)
```

```
EOF

        rm $FICH_STDOUT
        rm $FICH_STDERR

        #Envio de email
        mail -s "Backup de $MAQUINA_ORIGEN $HORA_COMIENZO" admins@arcos.inf.uc3m.es
                                                                < $FICH_LOG_SALIDA
else
        echo "Debe introducir un nombre de maquina, ejemplo: backup_raiz.sh piojito"
fi
```

8.4.31. Fichero *servicio_dns.sh*

```
#!/bin/sh
# Script que realiza una copia de seguridad del directorio de configuracion del dns

#set -x
SERVICIO=dns
MAQUINA_SERVICIO=piojito
DIR_ORIGEN=/etc/bind

DIR_DESTINO=/mnt/sistema/servicios/$SERVICIO
HORA_COMIENZO=$(date +%d_%B_%H-%M)
OPCIONES_RSYNC="-aud4z --stats --delete --backup --backup-dir=$DIR_DESTINO/$HORA_COMIENZO"


FICH_STDOUT=/root/temp/$SERVICIO-$HORA_COMIENZO-stdout.txt
FICH_STDERR=/root/temp/$SERVICIO-$HORA_COMIENZO-stderr.txt
FICH_LOG_SALIDA=/root/backup/salidas/servicio_$SERVICIO/$HORA_COMIENZO

/root/scripts/conectividad.sh $MAQUINA_SERVICIO
CONECTIVIDAD=$?

touch $FICH_STDOUT
touch $FICH_STDERR

if [ $CONECTIVIDAD -eq 0 ]; then

        nice -n 15 rsync $OPCIONES_RSYNC $MAQUINA_SERVICIO:$DIR_ORIGEN
                                        $DIR_DESTINO 1>> $FICH_STDOUT 2>> $FICH_STDERR

fi


HORA_FIN=$(date +%d_%B_%H-%M)
#Generando fichero de salida
```

```
cat > $FICH_LOG_SALIDA <<EOF
HORA DE COMIENZO=$HORA_COMIENZO
HORA FIN=$HORA_FIN
CONEXION CON $MAQUINA_SERVICIO=$(if [ $CONECTIVIDAD -eq 0 ]; then echo "OK";
                                                    else echo "FALLO"; fi)
SALIDA ESTANDAR
$(cat $FICH_STDOUT)

SALIDA ERROR
$(cat $FICH_STDERR)

EOF

rm $FICH_STDOUT
rm $FICH_STDERR

#Envio de email
mail -s "Backup de $SERVICIO $HORA_COMIENZO" admins@arcos.inf.uc3m.es < $FICH_LOG_SALIDA
```

8.4.32. Fichero *servicio_ldap.sh*

```
#!/bin/sh
# Script que realiza una copia de seguridad del directorio ldap.

#set -x
SERVICIO=ldap
MAQUINA_SERVICIO=piolin
DIR_ORIGEN=/var/lib/ldap

DIR_DESTINO=/mnt/sistema/servicios/$SERVICIO
HORA_COMIENZO=$(date +%d_%B_%H-%M)
FICH_DESTINO=fisico_$HORA_COMIENZO.bz2
FICH_DESTINO2=logico_$HORA_COMIENZO.ldiff

FICH_STDOUT=/root/temp/$SERVICIO-$HORA_COMIENZO-stdout.txt
FICH_STDERR=/root/temp/$SERVICIO-$HORA_COMIENZO-stderr.txt
FICH_LOG_SALIDA=/root/backup/salidas/servicio_$SERVICIO/$HORA_COMIENZO

/root/scripts/conectividad.sh $MAQUINA_SERVICIO
CONECTIVIDAD=$?

touch $FICH_STDOUT
touch $FICH_STDERR

if [ $CONECTIVIDAD -eq 0 ]; then

        ssh $MAQUINA_SERVICIO "tar -jcf /root/temp/$SERVICIO.bz2 $DIR_ORIGEN" \
                                        1>> $FICH_STDOUT 2>> $FICH_STDERR
```

```
        scp $MAQUINA_SERVICIO:/root/temp/$SERVICIO.bz2 $DIR_DESTINO/$FICH_DESTINO \
                                        1>> $FICH_STDOUT 2>> $FICH_STDERR
        ssh $MAQUINA_SERVICIO "rm /root/temp/$SERVICIO.bz2" \
                                      1>> $FICH_STDOUT 2>> $FICH_STDERR
        ssh $MAQUINA_SERVICIO "/usr/sbin/slapcat > /root/temp/$FICH_DESTINO2" \
                                        1>> $FICH_STDOUT 2>> $FICH_STDERR
        scp $MAQUINA_SERVICIO:/root/temp/$FICH_DESTINO2 $DIR_DESTINO/$FICH_DESTINO2 \
                                        1>> $FICH_STDOUT 2>> $FICH_STDERR
        ssh $MAQUINA_SERVICIO "rm  /root/temp/$FICH_DESTINO2" \
                                        1>> $FICH_STDOUT 2>> $FICH_STDERR
fi


HORA_FIN=$(date +%d_%B_%H-%M)
#Generando fichero de salida

cat > $FICH_LOG_SALIDA <<EOF
HORA DE COMIENZO=$HORA_COMIENZO
HORA FIN=$HORA_FIN
CONEXION CON $MAQUINA_SERVICIO=$(if [ $CONECTIVIDAD -eq 0 ]; then echo "OK";
                                                                 else echo "FALLO"; fi)
SALIDA ESTANDAR
$(cat $FICH_STDOUT)

SALIDA ERROR
$(cat $FICH_STDERR)

EOF

rm $FICH_STDOUT
rm $FICH_STDERR

#Envio de email
mail -s "Backup de $SERVICIO $HORA_COMIENZO" admins@arcos.inf.uc3m.es < $FICH_LOG_SALIDA
```

8.4.33. Fichero *servicio_mysql.sh*

```
#!/bin/sh
# Script que realiza una copia de seguridad de la base de datos mysql

#set -x
SERVICIO=mysql
MAQUINA_SERVICIO=piojito
DIR_ORIGEN=/var/lib/mysql

DIR_DESTINO=/mnt/sistema/servicios/$SERVICIO
HORA_COMIENZO=$(date +%d_%B_%H-%M)
FICH_DESTINO=fisico_$HORA_COMIENZO.bz2
FICH_DESTINO2=logico_$HORA_COMIENZO.sql
```

```
FICH_STDOUT=/root/temp/$SERVICIO-$HORA_COMIENZO-stdout.txt
FICH_STDERR=/root/temp/$SERVICIO-$HORA_COMIENZO-stderr.txt
FICH_LOG_SALIDA=/root/backup/salidas/servicio_$SERVICIO/$HORA_COMIENZO

/root/scripts/conectividad.sh $MAQUINA_SERVICIO
CONECTIVIDAD=$?

touch $FICH_STDOUT
touch $FICH_STDERR

if [ $CONECTIVIDAD -eq 0 ]; then

        ssh $MAQUINA_SERVICIO "tar -jcf /root/temp/$SERVICIO.bz2 $DIR_ORIGEN"
                                                        1>> $FICH_STDOUT 2>> $FICH_STDERR
        scp $MAQUINA_SERVICIO:/root/temp/$SERVICIO.bz2 $DIR_DESTINO/$FICH_DESTINO
                                                        1>> $FICH_STDOUT 2>> $FICH_STDERR
        ssh $MAQUINA_SERVICIO "rm /root/temp/$SERVICIO.bz2" 1>> $FICH_STDOUT 2>> $FICH_STDERR
        ssh $MAQUINA_SERVICIO "mysqldump -A -a -c --add-drop-table -q -Q -u backup
           --password=\"XXXXXXXX\" > /root/temp/tablas.sql" 1>> $FICH_STDOUT 2>> $FICH_STDERR
        scp $MAQUINA_SERVICIO:/root/temp/tablas.sql $DIR_DESTINO/$FICH_DESTINO2
                                                        1>> $FICH_STDOUT 2>> $FICH_STDERR
        ssh $MAQUINA_SERVICIO "rm  /root/temp/tablas.sql" 1>> $FICH_STDOUT 2>> $FICH_STDERR
fi


HORA_FIN=$(date +%d_%B_%H-%M)
#Generando fichero de salida

cat > $FICH_LOG_SALIDA <<EOF
HORA DE COMIENZO=$HORA_COMIENZO
HORA FIN=$HORA_FIN
CONEXION CON $MAQUINA_SERVICIO=$(if [ $CONECTIVIDAD -eq 0 ]; then echo "OK";
                                                                else echo "FALLO"; fi)
SALIDA ESTANDAR
$(cat $FICH_STDOUT)

SALIDA ERROR
$(cat $FICH_STDERR)

EOF

rm $FICH_STDOUT
rm $FICH_STDERR

#Envio de email
mail -s "Backup de $SERVICIO $HORA_COMIENZO" admins@arcos.inf.uc3m.es < $FICH_LOG_SALIDA
```

8.4.34. Fichero *backup_lucas_a_boyerito_usuarios.sh*

```
#!/bin/sh
#Script que realiza un backup completo de las cuentas en $MAQUINA_DESTINO

#set -x
MAQUINA_DESTINO=boyerito

/root/scripts/conectividad.sh $MAQUINA_DESTINO
CONECTIVIDAD=$?
if [ $CONECTIVIDAD -eq 0 ]; then
        DIR_ORIGEN=/mnt/usuarios

        DIR_DESTINO=$MAQUINA_DESTINO:/mnt/md0/
        HORA_COMIENZO=$(date +%d_%B_%H-%M)

        FICH_STDOUT="/root/temp/$MAQUINA_DESTINO-$HORA_COMIENZO-stdout.txt"
        FICH_STDERR="/root/temp/$MAQUINA_DESTINO-$HORA_COMIENZO-stderr.txt"
        FICH_LOG_SALIDA=/root/backup/salidas/$MAQUINA_DESTINO/cuentas_$HORA_COMIENZO

        OPCIONES_RSYNC="-aud4z --exclude-from=/root/backup/exclude_raiz --stats --delete"

        touch $FICH_STDOUT
        touch $FICH_STDERR
        ssh $MAQUINA_DESTINO mount /mnt/md0
        DIR_MONTADO=$(ssh $MAQUINA_DESTINO df |grep /mnt|wc|awk '{print $1}')

        if [ $DIR_MONTADO -eq 1 ]; then

                nice -n 15 rsync $OPCIONES_RSYNC $DIR_ORIGEN $DIR_DESTINO
                                                        1>> $FICH_STDOUT 2>> $FICH_STDERR
                ssh $MAQUINA_DESTINO umount /mnt/md0
        fi


        HORA_FIN=$(date +%d_%B_%H-%M)
        #Generando fichero de salida

        cat > $FICH_LOG_SALIDA <<EOF
HORA DE COMIENZO=$HORA_COMIENZO
HORA FIN=$HORA_FIN
CONEXION CON $MAQUINA_DESTINO=$(if [ $CONECTIVIDAD -eq 0 ]; then echo "OK";
                                                        else echo "FALLO"; fi)
SALIDA ESTANDAR
$(cat $FICH_STDOUT)

SALIDA ERROR
$(cat $FICH_STDERR)
```

```
EOF

        rm $FICH_STDOUT
        rm $FICH_STDERR

        #Envio de email
        mail -s "Backup de usuarios en $MAQUINA_DESTINO $HORA_COMIENZO"
                                admins@arcos.inf.uc3m.es < $FICH_LOG_SALIDA

else
        mail -s "No se ha podido conectar con $MAQUINA_DESTINO"
                                 admins@arcos.inf.uc3m.es < /dev/null
fi
```

8.4.35. Fichero *borrado_rotacion.sh*

```
#!/bin/sh
# Script para borrado de copias de seguridad según la fecha
# Francisco Olcina Grande 25/09/2006

DIR_MAQUINAS="/mnt/sistema/maquinas"
DIR_SERVICIOS="/mnt/sistema/servicios"
MAQUINAS="donald daisy lucas piojito piolin caponata"
SERVICIOS="mysql ldap dns"
EXCLUDE="raiz bind"

ITEM=$1
DIAS=$2

HORA_COMIENZO=$(date +%d_%B_%H-%M)
FICH_LOG_SALIDA=/root/backup/salidas/borrado/$ITEM-$HORA_COMIENZO


VALIDO=0
RESULTADO=$(echo $MAQUINAS| grep $ITEM)

if [ $(echo $?) -eq 0 ]; then
        VALIDO=1
        DIR_ITEM=$DIR_MAQUINAS/$ITEM
else
        RESULTADO=$(echo $SERVICIOS| grep $ITEM)
        if [ $(echo $?) -eq 0 ]; then
                VALIDO=1
                DIR_ITEM=$DIR_SERVICIOS/$ITEM
        fi
fi

if [ $VALIDO -eq 1 ]; then
```

```
        LISTA=$(find $DIR_ITEM -ctime +$DIAS -maxdepth 1)
        echo "Borrando de $ITEM los directorios:" >> $FICH_LOG_SALIDA

        for i in $LISTA; do
                EXCLUIR=0
                for e in $EXCLUDE; do
                        COMPROBACION=$(echo $i | grep $e)
                        if [ $(echo $?) -eq 0 ]; then
                                EXCLUIR=1
                        fi
                done

                if [ $EXCLUIR -eq 0 ]; then
                        echo "$i" >> $FICH_LOG_SALIDA
                        rm -fr $i 1>> $FICH_LOG_SALIDA 2>>$FICH_LOG_SALIDA
                fi
        done

fi

mail -s "Borrado de $ITEM $HORA_COMIENZO" admins@arcos.inf.uc3m.es < $FICH_LOG_SALIDA
```

8.4.36. Fichero *backup_donald_a_daisy_usuarios.sh*

```
#!/bin/sh
#Script que realiza un backup completo de las cuentas en $MAQUINA_DESTINO

set -x
MAQUINA_DESTINO=daisy

/root/scripts/conectividad.sh $MAQUINA_DESTINO
CONECTIVIDAD=$?
if [ $CONECTIVIDAD -eq 0 ]; then
        DIR_ORIGEN=/mnt/usuarios

        DIR_DESTINO=$MAQUINA_DESTINO:/mnt/usuarios/
        HORA_COMIENZO=$(date +%d_%B_%H-%M)

        FICH_STDOUT="/root/temp/$MAQUINA_DESTINO-$HORA_COMIENZO-stdout.txt"
        FICH_STDERR="/root/temp/$MAQUINA_DESTINO-$HORA_COMIENZO-stderr.txt"
        FICH_LOG_SALIDA=/root/backup/salidas/$MAQUINA_DESTINO/cuentas_$HORA_COMIENZO

        OPCIONES_RSYNC="-aud4z --exclude-from=/root/backup/exclude_raiz --stats --delete"

        touch $FICH_STDOUT
        touch $FICH_STDERR
        ssh $MAQUINA_DESTINO mount /mnt/usuarios
        DIR_MONTADO=$(ssh $MAQUINA_DESTINO df |grep /mnt|wc|awk '{print $1}')
```

```
        if [ $DIR_MONTADO -eq 1 ]; then
                mount /mnt/usuarios
                nice -n 15 rsync $OPCIONES_RSYNC $DIR_ORIGEN $DIR_DESTINO
                                                    1>> $FICH_STDOUT 2>> $FICH_STDERR
                ssh $MAQUINA_DESTINO umount /mnt/usuarios
                umount /mnt/usuarios
        fi


        HORA_FIN=$(date +%d_%B_%H-%M)
        #Generando fichero de salida

        cat > $FICH_LOG_SALIDA <<EOF
HORA DE COMIENZO=$HORA_COMIENZO
HORA FIN=$HORA_FIN
CONEXION CON $MAQUINA_DESTINO=$(if [ $CONECTIVIDAD -eq 0 ]; then echo "OK";
                                                                  else echo "FALLO"; fi)
SALIDA ESTANDAR
$(cat $FICH_STDOUT)

SALIDA ERROR
$(cat $FICH_STDERR)

EOF

        rm $FICH_STDOUT
        rm $FICH_STDERR

        #Envio de email
        mail -s "Backup de usuarios en $MAQUINA_DESTINO $HORA_COMIENZO"
                                            admins@arcos.inf.uc3m.es < $FICH_LOG_SALIDA

else
        mail -s "No se ha podido conectar con $MAQUINA_DESTINO"
                                            admins@arcos.inf.uc3m.es < /dev/null
fi
```

8.4.37. Fichero *durep.sh*

```
#!/bin/sh
#set -x


U_HOME=/mnt/home/
U_MAIL=/mnt/mail/
U_WEB=/mnt/web/
U_BACKUP=/mnt/backup/
```

```
#
mkdir /tmp/durep
durep -wd 2 -q -w /tmp/durep -x $U_HOME
scp -r /tmp/durep/* web@ssh:/mnt/home/otros/web/private_html/stats_disk_home/
rm -fr /tmp/durep

#
mkdir /tmp/durep
durep -wd 3 -q -w /tmp/durep -x $U_MAIL
scp -r /tmp/durep/* web@ssh:/mnt/home/otros/web/private_html/stats_disk_mail/
rm -fr /tmp/durep

#
mkdir /tmp/durep
durep -wd 3 -q -w /tmp/durep -x $U_WEB
scp -r /tmp/durep/* web@ssh:/mnt/home/otros/web/private_html/stats_disk_web/
rm -fr /tmp/durep

#
mkdir /tmp/durep
durep -wd 1 -q -w /tmp/durep -x $U_BACKUP
scp -r /tmp/durep/* web@ssh:/mnt/home/otros/web/private_html/stats_disk_backup/
rm -fr /tmp/durep
```

8.4.38. Fichero */usr/local/sbin/webalizer.sh*

```
 #!/bin/sh
set -x


# mkdir
rm -fr /tmp/stats_apache/
scp -r web@ssh:~/private_html/stats_apache /tmp/

# webalizer
cd /var/log/apache

/usr/sbin/logresolve2 < arcos.inf.uc3m.es-access_log >
                                    arcos.inf.uc3m.es-access_log.resolved
/usr/sbin/logresolve2 < arcos.inf.uc3m.es-error_log  >
                                    arcos.inf.uc3m.es-error_log.resolved
/usr/bin/webalizer -c /etc/webalizer-arcos.conf

/usr/sbin/logresolve2 < jornadas.arcos.inf.uc3m.es-access_log >
                                   jornadas.arcos.inf.uc3m.es-access_log.resolved
/usr/sbin/logresolve2 < jornadas.arcos.inf.uc3m.es-error_log  >
                                   jornadas.arcos.inf.uc3m.es-error_log.resolved
/usr/bin/webalizer -c /etc/webalizer-jornadas.conf
```

```
/usr/bin/webalizer -c /etc/webalizer-magsi.conf

/usr/bin/webalizer -c /etc/webalizer-mimpi.conf

/usr/bin/webalizer -c /etc/webalizer-winpfs.conf

/usr/bin/webalizer -c /etc/webalizer-xpn.conf


# copy and remove
scp -r /tmp/stats_apache/  web@ssh:~/private_html/
rm -fr /tmp/stats_apache/
```

8.4.39. Fichero */etc/webalizer-arcos.conf*

```
LogFile         /var/log/apache/arcos.inf.uc3m.es-access_log.resolved

OutputDir       /tmp/stats_apache/arcos/

Incremental     yes

ReportTitle     Uso del servidor Web

HostName        arcos.inf.uc3m.es

HideSite        *piojito

HideReferrer    piojito/

HideReferrer    Direct Request

HideURL         *.gif
HideURL         *.GIF
HideURL         *.jpg
HideURL         *.JPG
HideURL         *.ra

GroupURL        /cgi-bin/*

IgnoreSite      localhost
IgnoreReferrer  localhost

MangleAgents    4
```

8.4.40. Fichero */usr/local/bin/freq.sh*

```
#!/bin/sh
```

```
# login_per_users
echo "<html>"                                          > /tmp/sync.$$.txt
echo "<p> </p>"                                  >> /tmp/sync.$$.txt
echo "<ul> "                                          >> /tmp/sync.$$.txt
echo "<pre>"                                          >> /tmp/sync.$$.txt
/usr/local/bin/freq  -d -g -i root | sed 's/#/<img width=\"35\"
               src=\"\/imagenes\/smile01.jpg\">/g' >> /tmp/sync.$$.txt
echo "</pre>"                                         >> /tmp/sync.$$.txt
echo "</ul> "                                         >> /tmp/sync.$$.txt
echo "</html>"                                        >> /tmp/sync.$$.txt

scp /tmp/sync.$$.txt web@ssh:~/private_html/stats_user/login_per_user.html
rm -fr /tmp/sync.$$.txt


# full
echo "<html>"                                          > /tmp/sync.$$.txt
echo "<p> </p>"                                  >> /tmp/sync.$$.txt
echo "<font color=navy>"                              >> /tmp/sync.$$.txt
echo "<ul> "                                          >> /tmp/sync.$$.txt
echo "<ul> "                                          >> /tmp/sync.$$.txt
echo "<pre>"                                          >> /tmp/sync.$$.txt
/usr/local/bin/freq -e -g | sed 's/
/g'     >> /tmp/sync.$$.txt        /<br>
echo "</pre>"                                         >> /tmp/sync.$$.txt
echo "</ul> "                                         >> /tmp/sync.$$.txt
echo "</ul> "                                         >> /tmp/sync.$$.txt
echo "</font>"                                        >> /tmp/sync.$$.txt
echo "</html>"                                        >> /tmp/sync.$$.txt

scp /tmp/sync.$$.txt web@ssh:~/private_html/stats_user/full.html
rm -fr /tmp/sync.$$.txt

# lastlog
echo "<html>"                                          > /tmp/sync.$$.txt
echo "<p> </p>"                                  >> /tmp/sync.$$.txt
echo "<pre>"                                          >> /tmp/sync.$$.txt
/usr/bin/lastlog | sed 's/^root.*/<b>&<\/b>/g'      >> /tmp/sync.$$.txt
echo "</pre>"                                         >> /tmp/sync.$$.txt
echo "</html>"                                        >> /tmp/sync.$$.txt

scp /tmp/sync.$$.txt web@ssh:~/private_html/stats_user/lastlog.html
rm -fr /tmp/sync.$$.txt
```

8.4.41. Fichero */etc/munin/munin.conf*

```
dbdir   /var/lib/munin
```

```
htmldir /var/lib/munin/html
logdir  /var/log/munin
rundir  /var/run/munin

tmpldir /etc/munin/templates

graph_period minute

[caponata.arcos.inf.uc3m.es]
    address caponata.arcos.inf.uc3m.es
    use_node_name yes

[piojito.arcos.inf.uc3m.es]
    address piojito.arcos.inf.uc3m.es
    use_node_name yes

[piolin.arcos.inf.uc3m.es]
    address piolin.arcos.inf.uc3m.es
    use_node_name yes

[lucas.arcos.inf.uc3m.es]
    address lucas.arcos.inf.uc3m.es
    use_node_name yes

[donald.arcos.inf.uc3m.es]
    address donald.arcos.inf.uc3m.es
    use_node_name yes

[daisy.arcos.inf.uc3m.es]
    address daisy.arcos.inf.uc3m.es
    use_node_name yes
```

8.4.42. Fichero */etc/munin/munin-node.conf*

```
log_level 4
log_file /var/log/munin/munin-node.log
port 4949
pid_file /var/run/munin/munin-node.pid
background 1
setseid 1

host *
user root
group root
setsid yes

ignore_file ~$
ignore_file \.bak$
ignore_file %$
```

```
ignore_file \.dpkg-(tmp|new|old|dist)$
ignore_file \.rpm(save|new)$

host_name piojito.arcos.inf.uc3m.es

allow ^127\.0\.0\.1$
allow ^163\.117\.148\.240$
```

8.4.43. Fichero */etc/denyhosts.conf*

```
SECURE_LOG = /var/log/auth.log

HOSTS_DENY = /etc/hosts.deny

PURGE_DENY =

BLOCK_SERVICE = ALL

DENY_THRESHOLD_INVALID = 10

DENY_THRESHOLD_VALID = 15

DENY_THRESHOLD_ROOT = 5

DENY_THRESHOLD_RESTRICTED = 1

WORK_DIR = /var/lib/denyhosts

SUSPICIOUS_LOGIN_REPORT_ALLOWED_HOSTS=YES

HOSTNAME_LOOKUP=YES

LOCK_FILE = /var/run/denyhosts.pid

ADMIN_EMAIL = root@localhost

SMTP_HOST = localhost
SMTP_PORT = 25

SMTP_FROM = DenyHosts <nobody@localhost>

SMTP_SUBJECT = DenyHosts Report

AGE_RESET_VALID=5d

AGE_RESET_ROOT=25d

AGE_RESET_RESTRICTED=25d
```

```
AGE_RESET_INVALID=10d

DAEMON_LOG = /var/log/denyhosts


DAEMON_SLEEP = 30s

DAEMON_PURGE = 1h
```

Bibliografía

[1] Changelog-2.6.19. http://www.kernel.org/pub/linux/kernel/v2.6/ChangeLog-2.6.19, Julio 2007.

[2] Comunidad ldap española. http://www.ldap-es.org/, Julio 2007.

[3] Freq - lastlog analyzer (analizador del fichero lastlog). http://www.bangmoney.org/projects/freq/, Septiembre 2007.

[4] Herramientas smbldap-tools - instalación. http://es.tldp.org/Tutoriales/doc-openldap-samba-cups-python/html/ldap+samba+cups+pykota.html, Septiembre 2007.

[5] Linux-vserver. http://es.wikipedia.org/wiki/Linux-VServer, Julio 2007.

[6] Monitorización de servidores con munin y monit. http://www.howtoforge.com/server_monitoring_monit_munin, Septiembre 2007.

[7] Qemu. http://es.wikipedia.org/wiki/Qemu, Julio 2007.

[8] Qemu. http://www.damnsmalllinux.org/wiki/index.php/Qemu, Julio 2007.

[9] Xen - funcionamiento. http://pepssss.blogspot.com/2007/04/arquitectura-de-xen-antes-de-empezar.html, Julio 2007.

[10] Xen 3, linux-magazine. https://www.linux-magazine.es/issue/23/016-020_Xen3LM23.crop.pdf, Julio 2007.

[11] Xen en debian etch. http://www.linuxsilo.net/articles/xen.html, Julio 2007.