

# Directrices IDA de migración a software de fuentes abiertas



netproject

Netproject Ltd.  
124 Middleton Road  
Morden, Surrey  
SM4 6RW

Tel.: +44 (0)208 715 0072  
Fax: +44 (0)208 715 7134  
Web:  
[www.netproject.com](http://www.netproject.com)

	<b><i>Versión</i></b>
<p><u>Traducción en lengua española realizada del texto en inglés, versión oficial de esta publicación titulada:</u>  <i>The IDA Open Source Migration Guidelines</i>            © European Communities 2003. Reproduction is authorised provided the source is acknowledged.</p> <p><u>Para la edición en español:</u>            © 2004, Comisión Europea y Ministerio de Administraciones Públicas, España. Se autoriza la reproducción de esta obra citando la fuente.            Publicada con la autorización de la Comisión Europea.            La Comisión Europea no es responsable de la calidad de la traducción en español y de su coherencia con el texto original.</p>	Versión 1

NIPO: 326-04-042-8

Las opiniones expresadas en este documento son exclusivamente las de los autores y no se pueden interpretar, en ningún caso, como reflejo de una postura oficial de la Comisión Europea.

La Comisión Europea no garantiza la precisión de la información incluida en este estudio, ni acepta responsabilidad alguna derivada del uso de la misma.

Toda referencia que aquí se haga a productos concretos, especificaciones, procesos o servicios mediante nombre comercial, marca registrada, fabricante o de otra manera, no constituye ni implica necesariamente su respaldo, recomendación ni preferencia por parte de la Comisión Europea.

Los autores han tenido el debido cuidado en conseguir, siempre que fue necesario, el oportuno permiso para usar parte de manuscritos como, por ejemplo, ilustraciones, mapas y gráficos que están protegidos por derechos de propiedad intelectual, del titular o titulares de tales derechos o de su respectivo representante legal.

## 0.1 Historia del Documento

Fecha	Versión	Autor	Cambios
11/02/03	0.1	S. Hnizdur	Borrador inicial
13/04/03	0.2	S. Hnizdur	Añadidos y reorganización
14/05/03	0.3	S. Hnizdur	Añadidos y reorganización
20/05/03	0.4	S. Hnizdur	Preparación borrador para reunión PMB
21/05/03	0.5	S. Hnizdur	Inclusión de comentarios sobre Vo.4
31/05/03	0.6	S. Hnizdur	Inclusión de comentarios sobre Vo.5
30/06/03	0.7	S. Hnizdur	Inclusión de comentarios sobre Vo.6 y aplicaciones servidor MS
16/07/03	0.8	S. Hnizdur	Inclusión de comentarios sobre Vo.7 y Escenario Unix
17/07/03	0.9	S. Hnizdur	Inclusión de comentarios sobre Vo.8 y aplicaciones escritorio Windows
10/08/03	0.91	S. Hnizdur	Inclusión de comentarios sobre Vo.9
27/08/03	0.92	C.P. Briscoe-Smith	Correcciones e inclusión de algunos comentarios sobre Vo.9
04/09/03	0.93	C.P. Briscoe-Smith	Correcciones
08/09/03	0.94	C.P. Briscoe-Smith	Más correcciones e inclusión de comentarios
14/09/03	0.95	S. Hnizdur	Reorganización tras reunión PMB
25/09/03	0.96	S. Hnizdur	Correcciones
07/10/03	0.97	S. Hnizdur	Documento final
10/10/03	1.0	S. Hnizdur	Documento versión 1

## 0.2 Distribución

netproject Ltd x1

Frequentous Consultants Ltd x1

Comisión Europea (para difusión a las Administraciones de los Estados miembros) x2

## 0.3 Términos registrados

En este documento se han utilizados las marcas registradas exclusivamente con fines de identificación. Los autores reconocen la propiedad de estas marcas registradas.

## 0.4 Copyright (derechos de autor)

© European Communities. Se autoriza su reproducción siempre que se cite la fuente.

## 0.5 ÍNDICE

1. Prefacio.....	12
1.1 Abreviaturas y terminología.....	12
1.2. Audiencia.....	12
1.3. Autores.....	12
1.4. Reconocimientos.....	12
2. Introducción.....	15
3. Resumen.....	16
4. Metodología.....	18
5. Migración: visión general.....	21
6. Temas humanos.....	26
7. Una vida más fácil.....	28
7.1. Introducir nuevas aplicaciones en un entorno familiar.....	28
7.2. Lo fácil primero.....	28
7.3. Mirar hacia adelante.....	28
8. Arquitectura de referencia.....	32
8.1. Arquitecturas genéricas.....	32
8.2. Arquitectura de Referencia.....	34
9. Grupos funcionales.....	36
9.1. Los grupos principales.....	36
9.1.1. Ofimática.....	36
9.1.2. Correo.....	36
9.1.3. Calendarios y herramientas colaborativas de trabajo en grupo.....	36
9.1.4. Acceso a la web y servicios.....	36
9.1.5. Gestión de documentos.....	36
9.1.6. Bases de datos.....	36
9.2. Los grupos secundarios.....	36
9.3. Consideraciones generales.....	37
10. El modelo de referencia - Resumen.....	38
10.1. El escritorio.....	38
10.1.1. Ofimática.....	39
10.1.2. Correo.....	39
10.1.3. Calendarios y herramientas colaborativas de trabajo en grupo.....	39
10.1.4. Acceso a la Web.....	39
10.1.5. Gestión de documentos.....	40

10.1.6. Bases de datos.....	40
10.2. Los servidores.....	40
10.2.1. Correo.....	40
10.2.2. Calendarios y herramientas colaborativas de trabajo en grupo.....	40
10.2.3. Acceso a la Web .....	40
10.2.4. Gestión de documentos.....	40
10.2.5. Bases de datos.....	40
11. Aplicaciones – Grupos principales.....	41
11.1. Ofimática.....	41
11.1.1. OpenOffice.org y StarOffice .....	41
11.1.2. Koffice.....	42
11.1.3. Gnome Office.....	43
11.2. Correo.....	43
11.2.1. MTA.....	43
11.2.2. Buzón.....	44
11.2.3. MUA.....	45
11.2.4. Antivirus.....	45
11.2.5. Otras herramientas.....	46
11.2.6. Problemas experimentados.....	46
11.3. Calendarios y herramientas colaborativas de trabajo en grupo.....	47
11.3.1. Calendarios y agendas personales.....	49
11.3.2. Calendarios de grupo.....	49
11.3.3. Organización de reuniones.....	49
11.3.4. Sincronización de PDA.....	49
11.4. Servicios de Web.....	49
11.4.1. Navegador .....	49
11.4.2. Servidores Web.....	50
11.4.3. Portal / Contenido.....	51
11.5. Gestión de documentos.....	51
11.5.1. Registro y recuperación.....	51
11.5.2. Trabajo en colaboración.....	52
11.6. Bases de datos.....	52
11.6.1. Bases de datos generales según aplicaciones.....	52
11.6.2. Bases de datos personales de entorno local o central.....	52
11.6.3. Conectividad de bases de datos.....	53
11.6.4. Rendimiento.....	53

12. Aplicaciones – Grupos secundarios.....	54
12.1. Sistema Operativo.....	54
12.2. Interfaz de usuario.....	55
12.2.1. Gestor de escritorio – aspecto visual y operacional.....	55
12.2.2. Idioma .....	55
12.3. Seguridad.....	55
12.3.1. Cifrado de los datos.....	55
12.3.1.1. Datos en tránsito.....	55
12.3.1.2. Datos en memoria.....	56
12.3.2. Autenticación.....	56
12.3.3. Autorización.....	56
12.3.4. Control de virus.....	56
12.3.5. Servidor proxy.....	56
12.3.6. Cortafuegos.....	56
12.3.7. Redes privadas virtuales (VPN).....	57
12.3.7.1. OpenVPN.....	57
12.3.7.2. FreeSWAN.....	57
12.3.7.3. CIPE.....	57
12.4. Gestión.....	57
12.4.1. Gestión de usuarios.....	58
12.4.2. Gestión de configuraciones.....	58
12.4.2.1. Mantenimiento manual de la configuración.....	58
12.4.2.2. Cfengine.....	58
12.4.2.3. Configurador del sistema.....	58
12.4.3. Gestión del software.....	59
12.4.3.1. Instalación del software.....	59
12.4.3.2. Mantenimiento del software.....	61
12.4.4. Gestión del hardware y supervisión del sistema.....	62
12.4.4.1. MRTG y Snmpd.....	62
12.4.4.2. Nagios.....	63
12.4.4.3. smartd.....	63
12.4.5. Gestión de la impresora.....	63
12.4.5.1. LPRng.....	63
12.4.5.2. Sistema Común de Impresión Unix.....	63
12.4.5.3. Kprint y GnomePrint.....	63
12.5. Copia de seguridad y recuperación.....	63

12.5.1. Dump y Restore.....	64
12.5.2. Amanda.....	64
12.6. Otros servicios.....	64
12.6.1. Servidores temporales.....	64
12.6.2. Servidores de infraestructuras de red.....	64
12.6.2.1. Encaminamiento.....	64
12.6.2.2. DNS.....	64
12.6.2.3. DHCP.....	65
12.6.3. Servidores de archivos.....	65
12.6.3.1. NFS.....	65
12.6.3.2. Samba.....	65
12.6.3.3. Netatalk.....	65
12.6.3.4. OpenAFS, CODA e Intermezzo.....	65
12.6.4. Servicios de directorio.....	66
12.6.5. Soporte de sistemas heredados.....	66
12.6.5.1. Emulación por terminal.....	66
12.6.5.2. Visualización remota.....	66
12.6.5.3. Emulación.....	66
13. Migración de aplicaciones – Visión general.....	67
13.1. Aplicaciones propietarias que tienen un equivalente OSS.....	67
13.2. Aplicaciones propietarias que funcionan en un entorno OSS.....	67
13.3. Software al que se puede acceder por visualización remota.....	67
13.4. Software que funcionará con un emulador.....	68
13.4.1. Emulación del hardware.....	68
13.4.2. Emulación del software.....	69
13.5. Software que se puede recompilar en OSS.....	70
14. Escenario 1 - Windows.....	72
14.1. Planificar la migración .....	72
14.2. Dominios.....	72
14.2.1. Modelo de "grupo de trabajo" Windows.....	72
14.2.2. Dominio NT de Windows.....	72
14.2.3. Dominio de Active Directory en Windows 2000.....	73
14.3. Vista general de las posibles rutas de migración.....	73
14.4. Temas generales.....	74
14.4.1. Nombre de usuario y palabras de paso.....	75
14.4.1.1. Nombres de usuario.....	75

14.4.1.2. Contraseñas.....	76
14.4.2. Servicios de autenticación.....	76
14.4.3. Archivos.....	76
14.4.3.1. Formato y contenido.....	76
14.4.3.2. Nombres de archivos .....	77
14.4.3.3. Acceso doble.....	78
14.5. Herramientas.....	79
14.5.1. Samba.....	79
14.5.2. OpenLDAP.....	79
14.5.3. NSS y PAM.....	80
14.5.4. Acceso a archivos SMBFS de GNU/Linux.....	80
14.5.5. Winbind.....	80
14.6. Migración del entorno de sistema operativo.....	81
14.6.1. Añadir servidores individuales de GNU/Linux a un Dominio Windows NT existente.....	81
14.6.2. Ejecutar escritorios GNU/Linux en dominios Windows NT.....	81
14.6.2.1. Configuración sencilla para pocas máquinas.....	81
14.6.2.2. Configuración elaborada para despliegues mayores .....	83
14.6.3. Ejecutar escritorios GNU/Linux en dominios de Directorio Activo.....	85
14.6.4. Cambiar PDC/BDC Windows NT por Samba+LDAP .....	85
14.6.5. Cambiar el Directorio Activo Windows 2000 por el LDAP.....	86
14.6.6. Ejecutar Infraestructura GNU/Linux paralela y transferir usuarios en grupos ...	87
14.6.6.1. Reemplazar todos los clientes de Windows por GNU/Linux.....	87
14.6.6.2. Conservar algunos clientes de Windows.....	88
14.7. Migración de aplicaciones del servidor.....	88
14.7.1. Servidores Web: migrar de IIS a Apache.....	88
14.7.1.1. Temas de migración.....	88
14.7.1.2. Migración de un sitio web estático.....	90
14.7.1.3. Una sencilla configuración WebDAV.....	91
14.7.2. Bases de datos: migrar de Access y SQL Server a MySQL o PostgreSQL.....	93
14.7.2.1. Migración de las bases de datos de Access.....	93
14.7.2.2. Migración de las bases de datos de SQL Server.....	94
14.7.2.3. Temas de migración de bases de datos.....	95
14.7.3. Programas de grupo: salirse de Exchange.....	95
14.7.3.1. Temas generales.....	95



14.7.3.2. Correo.....	96
14.7.3.3. Carpeta de direcciones.....	96
14.7.3.4. Calendario.....	96
14.8. Migración de aplicaciones del escritorio a OSS.....	97
14.8.1. Ofimática.....	97
14.8.1.1. Conversión de documentos.....	97
14.8.1.2. Conversión de plantillas.....	97
14.8.1.3. Conversión de macros.....	97
14.8.1.4. Procesadores de textos.....	97
14.8.1.5. Autoedición.....	98
14.8.1.6. Hojas de cálculo.....	98
14.8.1.7. Gráficos para presentaciones.....	99
14.8.1.8. Manipulación de imágenes y gráficos.....	99
14.8.1.9. Generar PDF.....	100
14.8.2. Correo .....	100
14.8.3. Calendarios y Programas de grupo.....	101
14.8.3.1. Calendarios.....	101
14.8.3.2. Gestión de contactos .....	101
14.8.4. Navegación en la web.....	102
14.8.5. Bases de datos personales.....	102
14.9. Migración de los servicios de impresión a OSS.....	103
14.9.1. El modelo de impresión de Windows.....	103
14.9.2. Unix y el modelo de impresión de GNU/Linux.....	103
14.9.3. Configurar un servicio de impresión basado en OSS.....	104
14.9.4. Imprimir desde clientes de Windows a impresoras adscritas a GNU/Linux...105	
14.9.4.1. Usando el protocolo Ipr.....	105
14.9.4.2. Usando impresoras compartidas.....	105
14.9.4.3. Usando configuración Point and Print (Señale e Imprima) .....	106
14.9.5. Imprimir esquemas de migración.....	106
14.9.6. Posibles problemas.....	107
14.9.7. Más información sobre la impresión.....	107
14.10. Aplicaciones heredadas.....	107
14.11. Protección frente a los virus.....	107
14.12. Referencias.....	108
15. Escenario 2 – Unix.....	109
16. Escenario 3 - Mainframe.....	111

17. Escenario 4 – Cliente ligero (thin client).....	112
Anexo A. Estudios de casos publicados.....	114
Anexo B. Wine.....	116
Anexo C. Sistemas de Correo.....	121
Anexo D. Software de Referencia para el Escritorio.....	129
Anexo E. Software de Referencia para el Servidor.....	135
Anexo F. Código para la instalación del escritorio.....	145
Anexo G. Glosario.....	151

## 1. Prefacio

### 1.1 Abreviaturas y terminología

Siempre que sea posible, la primera vez que aparece una abreviatura se incluye también su versión amplia. En el Anexo G se da un glosario de términos. Cuando un término del glosario se presenta por primera vez, se indicará de esta forma: **Glosario**.

Las expresiones **Open Source Software** (Software Abierto o de Fuentes abiertas) y **Free Software** (Software Libre) tienen ambas sus defensores. En este documento utilizamos software de fuentes abiertas (*Open Source Software* o bien **OSS**) y al hacerlo queremos que se entienda que el software descrito tiene las características implícitas en las dos expresiones (*Open Source Software* y *Free Software*). En las siguientes direcciones hay más información sobre las mismas: <http://www.gnu.org/philosophy/categories.html/> y <http://www.opensource.org/>.

Los nombres de productos se presentarán así: *Nombre de Producto*.

Los términos del ámbito de los sistemas operativos como, por ejemplo, nombres de archivos, se presentarán así: **Nombre de archivo**.

El código de programa, aparecerá así: `Code`.

### 1.2. Audiencia

Este documento va dirigido a los gestores de TI de las administraciones públicas europeas. El término **Administración** se usa a lo largo de las directrices para indicar una administración pública europea y el de **Administradores** se refiere a este grupo de personas.

### 1.3. Autores

Este informe ha sido realizado por **netproject Ltd.** en colaboración con **Frequentous Consultants Ltd.** El contenido fue elaborado por varios consultores, entre los que figuran los siguientes:

- Steve Hnizdur
- Keith Matthews
- Eddie Bleasdale
- Alain Williams
- Andrew Findlay
- Sean Atkinson
- Charles Briscoe-Smith

### 1.4. Reconocimientos

**netproject** quiere expresar su agradecimiento por su ayuda a los siguientes:

- En particular, **netproject** desea agradecer a los numerosos miembros del Comité de Gestión del Proyecto los vivos e informativos debates mantenidos mensualmente.
- Thomas Krupp y Heiko Thede (Mecklenburg-Verpommern)
- Frank Müller (Landesrechnungshof Mecklenburg-Verpommern)
- Mru Patel (Sun Microsystems)
- Graham Taylor (Open Forum Europe)
- Richard Heggs (Nottingham City Council)
- Andy Lack (City University London)
- David Amos (BSS Group)
- Steven Pennant (London Borough of Newham)
- Andy Trevor (Total Solutions)
- Geoff Gunton (Northern Rock)

**1ª PARTE**  
**Introducción**  
**y Sumario**

## 2. Introducción

El objeto de estas directrices es doble:

1. Ayudar a los Administradores a decidir si se debe emprender una migración a OSS.
2. Describir en lenguaje técnico claro cómo se debería llevar a cabo dicha migración.

Se pretende que las directrices tengan un uso práctico por parte de los Administradores y por ello deben ser oportunas y precisas a la vez que accesibles y comprensibles. No son un manual de referencias técnicas detalladas. Su estructura persigue que se puedan introducir modificaciones con facilidad a medida que se vaya adquiriendo experiencia por las Administraciones y a medida que se vayan introduciendo cambios en los productos disponibles.

***Para alcanzar estos objetivos es imperativo que el contenido esté siempre actualizado y que se elimine cualquier imprecisión. Para ello, encarecemos a los lectores que nos hagan los comentarios y aportaciones que consideren oportunas sobre cualquier parte de las directrices. Rogamos envíen sus comentarios a [gposs@cec.eu.int](mailto:gposs@cec.eu.int).***

Las directrices no hablan sobre el OSS en general ni de los meritos de las diversas licencias. Esto puede encontrarse, junto con mucha más información, en los siguientes documentos IDA:

1. El informe sobre los hechos del OSS:  
<http://europa.eu.int/ISPO/ida/export/files/en/840.pdf>
2. El informe sobre el uso del OSS:  
<http://europa.eu.int/ISPO/ida/export/files/en/837.pdf>
3. El informe sobre la estructura del mercado y los temas relativos a la contratación pública:  
<http://europa.eu.int/ISPO/ida/export/files/en/835.pdf>

(Los tres documentos anteriores se pueden encontrar en:

<http://europa.eu.int/ISPO/ida/jsps/index.jsp?fuseAction=showDocument&parent=crossreference&documentID=333>

en otros formatos además de en PDF).

La “Agence pour le Développement de l’Administration Électronique” (ADAE), francesa, ha elaborado una buena guía sobre licencias, que también está disponible en inglés en:

[http://www.atika.gouv.fr/pages/documents/fiche.php?od=1450&od\\_chapitre=8&id\\_theme=55&detype=0](http://www.atika.gouv.fr/pages/documents/fiche.php?od=1450&od_chapitre=8&id_theme=55&detype=0)

### 3. Resumen

Estas directrices van dirigidas a gestores y profesionales de Tecnología de la Información (TI) que estén planificando o ejecutando una migración a software de fuentes abiertas (OSS). Se basan en la experiencia práctica de los autores y en el contenido de un número limitado de experiencias públicamente accesibles. Estas directrices se han validado en el proceso de migración a OSS realizado por el Tribunal de Cuentas de Schwerin en el *lander* alemán de Mecklenberg Vorpommern.

Muchas son las razones de los Administradores para emprender una migración a OSS. Entre ellas se pueden citar las siguientes: la necesidad de normas abiertas para la administración electrónica, el nivel de seguridad que proporciona el OSS, la eliminación de las migraciones forzadas y el coste del OSS. Todas estas ventajas redundan en un coste mucho menor de la TI.

Estas directrices recomiendan:

- antes de comenzar tener una idea clara de las razones de la migración;
- asegurarse de que se cuenta con apoyo activo para el cambio por parte del personal de TI y los usuarios;
- asegurarse de que hay un defensor del cambio, mejor si está en el puesto más alto de la entidad;
- adquirir experiencia y establecer relaciones con el movimiento OSS;
- comenzar con sistemas no críticos;
- asegurarse de que cada etapa de la migración es factible.

La migración de sistemas de TI ofrece la oportunidad de rediseñarlos para satisfacer las nuevas expectativas puestas en ellos. Entre las cuestiones a tratar se incluyen:

- cómo garantizar la interoperabilidad de los sistemas;
- cómo dar soporte a los usuarios móviles;
- cómo identificar con seguridad a los usuarios remotos;
- cómo crear sistemas que son gestionables.

Sobre todo cómo garantizar que la seguridad se ha incorporado desde el inicio y no es algo que ha surgido después.

En el entorno servidor, el OSS se entiende bien y se emplea mucho. La migración de los servidores a OSS se puede hacer generalmente sin que afecte negativamente a los usuarios. Es por donde se suele empezar.

Para la mayoría de las empresas, incluir el OSS en el escritorio implica un ahorro de costes máximo. Al ser transferidas al escritorio, las nuevas aplicaciones de OSS habrán de interactuar con las ya existentes. En particular, se debe controlar el modo en que los servicios de calendario de grupo interactúan tanto en puestos OSS como propietarios.

Al reemplazar el software propietario de automatización de oficinas se deben comprobar las plantillas para ver si producen los resultados correctos. Hay que volver a escribir las macros, preferentemente como *scripts*. Las aplicaciones que no tienen equivalentes de OSS

se pueden ejecutar como clientes ligeros. Con el tiempo las aplicaciones del escritorio se pueden reemplazar por equivalentes de tipo OSS.

Aunque las directrices contemplan un cambio completo a OSS, lo más probable es que se construya un entorno heterogéneo especialmente teniendo en cuenta que una migración de miles de escritorios lleva su tiempo. Es probable además que se produzca la mezcla de aplicaciones propietarias y OSS pues no siempre será posible o adecuado sustituir las aplicaciones por otras OSS. De momento esto es realmente cierto en el caso de la sustitución de la función de trabajo colaborativo en grupos de Exchange de Microsoft. Sin embargo, hay bastantes aplicaciones de OSS de calidad suficiente para emprender la migración.

Es importante asegurarse de que las decisiones tomadas ahora, incluso si no tienen que ver directamente con una migración, no va a atar a una Administración a protocolos y formatos de archivos propietarios.

El OSS es una tecnología rompedora. Permite un cambio fundamental en el modo en que las entidades prestan servicios de TI. Es el paso de una industria basada en el producto a otra basada en el servicio. La instalación del software de OSS no cuesta nada. La cuestión es dónde conseguir apoyo. Hay una serie de empresas que dan soporte a terceros y también están los vendedores de distribuciones. No obstante, si su actitud ante la TI es “¿A quién pido cuentas cuando algo vaya mal?”, quizás el OSS no sea para usted. Es necesario entender la dinámica del funcionamiento del OSS. Resulta aconsejable saber cómo relacionarse con la comunidad OSS.



## 4. Metodología

Cualquier ejercicio de migración debe incluir, en general, lo siguiente:

1. Una fase de definición del proyecto y de recopilación de datos, en la que se contemplan:
  - A. La descripción del conjunto de condiciones iniciales relevantes consistentes, por ejemplo, en:
    - a) arquitectura o arquitecturas de los sistemas,
    - b) aplicaciones y sus datos asociados,
    - c) protocolos y normas empleados,
    - d) hardware,
    - e) el entorno físico, como el ancho de banda de la red, la ubicación,
    - f) los requisitos sociales como el idioma o idiomas y la capacitación del personal;
  - B. Un conjunto de condiciones finales con el mismo detalle,
  - C. Una descripción de cómo llegar de las condiciones iniciales a las condiciones finales;
2. Una justificación de la migración, incluido el coste asociado a la misma;
3. Una o más fases piloto preparadas para probar si el plan y la justificación funcionan. Los datos de estas fases piloto pueden luego alimentar el modelo de costes usado en el plan,
4. Despliegue del plan
5. Seguimiento de la experiencia real en relación con el plan.

En el punto 1 anterior se define lo que en estas directrices se denomina **Escenario** y las directrices describen cómo realizar la migración a OSS en tales circunstancias.

Sin embargo, para que las directrices resulten comprensibles y útiles en la práctica, hay que pensar en cierta simplificación pues de lo contrario el número total de combinaciones posibles sería irrealizable.

Hemos elegido uno de los muchos entornos finales diferentes (punto 1B) y hemos simplificado la descripción de los entornos iniciales (punto 1<sup>a</sup>). Del entorno final se habla en el apartado 8.2. Con este entorno final estándar asumido, un Escenario se define en referencia al entorno inicial simplificado y al camino de migración desde él al entorno final.

Se pueden añadir nuevos capítulos a este documento si es necesario en el futuro. A partir del capítulo 14, cada uno da una descripción bastante detallada de lo que es un Escenario, explica cómo realizar la migración al entorno final, sin olvidar el tratamiento de las migraciones parciales. Se debería adaptar cada capítulo a medida que se va adquiriendo experiencia con las migraciones reales.

Además, como ayuda para entender el punto 2 anterior, hay una hoja de cálculo en la que el coste de cada Escenario se puede comparar con el coste del entorno final junto con el coste de la migración.

La información disponible en los casos publicados ha sido bastante limitada. Sólo se encontró un reducido número de dichos estudios (en el Anexo A se da una lista de ellos)

que no aportan otros datos aparte de un artículo de prensa general. Esto significa que en gran medida las directrices se basan en la experiencia de **netproject** y sus asesores así como en sus conversaciones con quienes han hecho la migración pero no han publicado los resultados obtenidos.

El enorme número de combinaciones diferentes de condiciones iniciales y finales, junto con los muy diferentes modos de pasar de unas a otras significa que no es posible que las directrices cubran todas las posibilidades. Las directrices deben entonces considerarse como indicativas de lo que se puede hacer más que como normas de lo que debería hacerse. Se han de usar como punto de partida en el proceso de migración. Y no se debe esperar que aporten una respuesta en todas las circunstancias.

Si bien se asume que la migración persigue un entorno totalmente OSS en la medida de lo posible, puede haber razones que justifiquen la necesidad o el mantenimiento de sistemas propietarios. También se trata la posibilidad de una migración parcial.

**2ª PARTE**  
**Directrices**  
**de Gestión**

## 5. Migración: visión general

Lo que hay que hacer para migrar desde un entorno propietario a un entorno OSS es en gran medida similar a cualquier otra migración, por ejemplo de *Windows NT* a *Windows 2000*. Incluso en este cambio dentro de un mismo vendedor no se debe asumir que los formatos de archivos, por ejemplo, serán transportables, por lo que habrá que hacer pruebas de funcionamiento antes de realizar cualquier cambio amplio. Todas las migraciones deben basarse en una cuidadosa planificación.

Estas directrices no pretenden ser un manual sobre la gestión de proyectos y se supone que la Administración tiene los suficientes medios y conocimientos para gestionar adecuadamente la migración. La descripción que damos a continuación solo pretende hacer hincapié en algunos puntos destacados de una migración a OSS.

El proceso de migración ideal debería consistir en las siguientes partes, y algunas de ellas pueden hacerse en paralelo como 2, 3 y 4.

Obsérvese que la información encontrada puede indicar que habrá que hacer modificaciones en el entorno actual antes de preparar una migración a OSS. Por ello las administraciones que no tengan planes inmediatos de migración, pero que quieran mantener la posibilidad de hacerlo, deberían tener sólo estándares abiertos de varios vendedores y valorar su infraestructura de cara a ello (véase también el Capítulo 7.3 “Mirar hacia delante”).

1. Crear un equipo con la capacitación y el respaldo de gestión adecuados. Es importante que se disponga de apoyo de gestión pues de lo contrario habrá resistencia a un cambio de la norma de sistemas propietarios. Este apoyo habrá de ser suficiente para permitir por lo menos la construcción de pilotos representativos, así habrá que elaborar un caso de negocio básico, y quizá uno más detallado después cuando se disponga de más datos.
2. Entender el entorno final, tanto el software OSS como la arquitectura básica (véase capítulo 8), junto con las diferentes opciones y posibilidades disponibles. Esto significa que hay que formar al personal, contratar personal o recurrir a consultores. Esto implica algunos costes iniciales y por ello es necesario disponer de respaldo de los responsables de la gestión. A veces existe la expectativa de que el software libre se puede entender y usar sin coste alguno. **Pero este no es el caso.**
3. La migración es una oportunidad de revisar la arquitectura de base así como el software de aplicaciones. La arquitectura recomendada en el Capítulo 8 se basa en el control centralizado y tiene ciertas ventajas de las que se habla en el mismo capítulo. Puede haber ciertos costes al hacer el cambio, y hay que tenerlos en cuenta.
4. Es muy importante entender bien en qué consiste el OSS. Hay algunos aspectos que hay que tener en cuenta antes de tomar ninguna decisión.
  - A. Hay que tener claro cuáles son las implicaciones de las licencias para OSS especialmente si se considera que la Administración va a distribuir los cambios de software. En la Introducción se encuentran más documentos sobre el tema.
  - B. Cuando hay varias opciones para una función (por ejemplo, hay por lo menos tres buenas hojas de cálculo de OSS) los Administradores han de entender los pros y los contras de cada producto.

- C. Se deben tener en cuenta las diferencias entre las distintas distribuciones. Algunas distribuciones están respaldadas por empresas comerciales que prestan su apoyo y correcciones. Algunas tienen características diferentes: *Gentoo*, por ejemplo, da una distribución basada en un **código fuente** que facilita a la Administración una adaptación del software para que satisfaga sus necesidades concretas. Todas estas diferencias han de ser valoradas antes de hacer cualquier elección.
- D. Los Administradores deben determinar qué nivel de apoyo es necesario. Se puede conseguir apoyo comercial de los creadores de la aplicación o la distribución si la suministran. Si no es así, hay terceros que pueden prestar ese apoyo ya que se dispone del código fuente y hay muchas empresas que dan ese apoyo.

Esta es una diferencia clara respecto al mercado de software propietario donde un apoyo detallado sólo lo facilitan las empresas que tienen el privilegio de acceder al código fuente. Y esto es importante si el vendedor propietario abandona el negocio sin revelar el código fuente.

Y si todo falla, la mayoría de las aplicaciones tienen listas de correo activas donde una pregunta o petición de ayuda recibirá la respuesta de alguien relacionado con la aplicación. La presencia de una lista de correo activa y una comunidad de usuarios suele ser uno de los criterios a tener en cuenta en primer lugar en la elección de los componentes de software.

- 5. Estudiar los sistemas existentes. Estos datos no solo serán necesarios para hacer la migración en sí, sino que muchos de ellos serán también muy necesarios para construir un modelo de coste total de propiedad para un caso concreto de negocio. Hay que compilar los inventarios de lo siguiente:

- A. Para cada aplicación usada:

- a. El nombre de la aplicación, el número de la versión y el punto de contacto para obtener respuesta a cualquier consulta.
- b. Cuántos usuarios necesitan acceder a la misma.
- c. Qué sistema operativo se está usando. Con qué sistemas operativos se puede ejecutar la aplicación, incluidos entornos como *Citrix*.
- d. Qué otras aplicaciones necesitan el cliente y el servidor para que la aplicación funcione.
- e. Qué hardware se necesita. En particular si se necesita algún hardware de perfil especial o no estándar.
- f. Qué protocolo utiliza para comunicarse con otras aplicaciones.
- g. Qué formatos de archivos necesita.
- h. Qué internacionalización y localización se necesita. Se pueden necesitar múltiples idiomas y monedas.

- B. Requisitos en cuanto a los datos. Esto habría de interpretarse en el sentido amplio que contempla, por ejemplo, el proceso de textos y hojas de cálculo, voz y audio e imagen así como bases de datos normales. En general, todo lo que un ordenador ha de procesar.

- a. ¿Cuáles son las condiciones de interfaz con otros sistemas o usuarios que no controle la Administración?

- b. ¿Qué requisitos hay para conservar los datos y procesarlos en el futuro? ¿Hay un depósito de datos heredados existentes que haya que respaldar? Si es así, ¿son necesarias aplicaciones especiales para ese respaldo?
  - i. Dividir los datos en las siguientes categorías:
  - ii. Datos que no es necesario conservar y se pueden eliminar. Eliminarlos.
  - iii. Datos que deben conservarse y están actualmente en formato abierto como PDF o Proscript, o que pueden traducirse fácilmente a uno de ellos. Se debe estudiar con cuidado el coste de esa traducción.
  - iv. Datos que deben conservarse pero que están en un formato cerrado propietario que no puede traducirse fácilmente a otro abierto. Estos datos pueden requerir la conservación de copias de la aplicación propietaria concreta. Se debe estudiar el coste de estas aplicaciones. El número necesario de copias de la aplicación se puede determinar según el grado de acceso a los mismos que se requiere. Por ejemplo, si se accede poco a los datos entonces bastará con una copia en una máquina central. También puede ser necesario conservar un hardware específico para ejecutar esas aplicaciones.

C. Requisitos de seguridad:

- a. ¿Cuál es el sistema actual para asignar los nombres de usuarios y las contraseñas? ¿Tienen los nombres de usuario una estructura y si es así cuál? ¿Cuál es la política para actualizar las contraseñas?
  - b. ¿Hay sistemas que requieran otra autenticación más allá que un simple nombre de usuario y una contraseña?
  - c. ¿Qué políticas tiene la Administración en relación con el uso de los ordenadores? Por ejemplo ¿hay restricciones sobre el uso de Internet y el correo electrónico?
  - d. ¿Hay disposiciones de seguridad que exijan el uso de un hardware o software concretos?
6. Elaborar un caso detallado de migración, que se basará en los datos recogidos según lo dicho anteriormente y que consistirá en los siguientes puntos:
- A. El coste del entorno existente en un período de tiempo razonable digamos de cinco años con supuestos adecuados a la Administración.
  - B. El coste de entornos alternativos y el coste de la migración a cada uno de ellos en el mismo período.
  - C. Los puntos fuertes y débiles del entorno actual y las distintas alternativas.
- La correspondiente hoja de cálculo será de ayuda para comparar costes.
7. Consultar a los usuarios. Explicar las razones que hay detrás de la migración y cómo les afectará. Estudie sus preocupaciones con seriedad y permítales que practiquen con la tecnología sin pérdida de tiempo. Cuanto antes se impliquen los usuarios mejor será. Esto puede ser un requisito legal en algunos países pero habría que hacerlo en todo caso para facilitar la introducción de lo que puede ser un cambio significativo en las prácticas de trabajo.

Crear una ventanilla de atención al cliente que pueda dar respuesta a las preocupaciones de los usuarios. Más adelante, cuando la migración esté configurada, podrá resolver los problemas y convertirse en un centro de excelencia y buenas prácticas. Crear un sitio de Intranet con una sección dedicada a “consejos y cómo se hace” que los propios usuarios puedan actualizar. Es importante que los usuarios sientan que forman parte y éste sitio a su vez puede proporcionarle a la ventanilla de atención una idea del tipo de problemas a los que se enfrentan los usuarios.

8. Suponiendo que el caso de ejemplo se ha hecho, comenzar con proyectos piloto a pequeña escala, de preferencia en un entorno auto-contenido con pocos usuarios. Esto facilitará, entre otras cosas:
  - A. Datos más ajustados de modelos de coste total de propiedad.
  - B. La reacción de los usuarios, que se puede emplear para facilitar la introducción a otros sistemas.
  - C. La validación o modificación de la arquitectura final y el caso de ejemplo.
9. Decidir sobre la velocidad del proceso de migración una vez iniciado. Estas son las principales alternativas:
  - A. **Big bang:** Todos los usuarios cambian del viejo sistema al nuevo el mismo día. En la práctica, esto significa programar el cambio en un fin de semana o fiesta nacional. La ventaja es que no se necesitan disposiciones de doble acceso y el personal no se va a encontrar yendo y viniendo de un sistema a otro. Entre las desventajas está el alto riesgo y la gran exigencia de recursos durante el cambio. Este esquema de migración sólo cabe en el caso de las pequeñas Administraciones.  
  
En todo caso, si es posible **HAY QUE EVITAR LA MIGRACIÓN DE TIPO BIG BANG.** Las migraciones “big bang” tendrán tantas variantes que controlar que casi siempre fallan. Y si lo hacen no parece probable que sea por un fallo del OSS sino de gestión.
  - B. **Transición por fases en grupos:** Se pasa a los usuarios del antiguo sistema al nuevo en grupos. Puede que los grupos funcionales completos se trasladen juntos para minimizar tener que compartir datos y los problemas de trabajo en el grupo. Se pueden contener los riesgos y gestionar los recursos eligiendo grupos del tamaño adecuado. También es posible hacer un cambio del hardware de los PC al mismo tiempo, reemplazando las máquinas en un grupo y luego instalando las sustituidas en lugar de las viejas máquinas del siguiente grupo.
  - C. **Transición de usuario a usuario:** Básicamente la misma opción de la transición en grupos, pero con un grupo compuesto por una sola persona. Ese método de “goteo” tiene escasos requisitos en cuanto a los recursos, pero no resulta eficaz ni apropiado para grandes Administraciones. Pero sí puede ser una buena manera de ejecutar los proyectos piloto.

Es probable que tanto los viejos como los nuevos sistemas tengan que funcionar “codo con codo” durante cierto tiempo. Es importante contar con una estrategia de transición que permita que ambos sistemas funcionen juntos, de manera que las actividades de producción se puedan continuar correctamente durante el período de transición. La sustitución de la máquina vieja puede llevar bastante tiempo (o no tener lugar), por lo que la coexistencia puede ser muy importante.

10. Extender la migración a toda la Administración. Esto implicará más formación de los usuarios y del personal técnico.

11. Supervisar la respuesta de los usuarios y tomar nota de los problemas que surjan. Algunas necesidades de los usuarios pueden ser tan poco claras que no se pueden prever ni descubrir durante los proyectos piloto. Hay que asegurarse de que se dispone de recursos suficientes para hacer frente a esas necesidades tras la transición.

De todas maneras, es posible que en cualquier momento se vea que la migración no es factible. Esto podría deberse, por ejemplo, a que hay aplicaciones críticas que no funcionan bien en el entorno OSS y el coste de rescribirlas es demasiado alto.



## 6. Temas humanos

Estas directrices no pretenden ser una guía de Gestión de Recursos Humanos, y las Administraciones ya se habrán encontrado anteriormente con estas cuestiones en otras áreas. Tendrán capacidad hacerles frente de manera amable y favorable y así el personal de Recursos Humanos debería implicarse desde el principio. La intención es simplemente resaltar el tipo de cuestiones que han surgido en otros sitios que han migrado a OSS.

Es muy importante que se consulte a todo el personal y que se le mantenga informado de lo que se va haciendo. Un modo de hacerlo es crear una intranet que se pueda mantener actualizada fácilmente y en la que haya una sección dedicada a las opiniones de los usuarios.

El acceso a la formación es muy importante. Algunos sitios permiten a los usuarios decidir por sí mismos si quieren asistir mientras que otros exigen la asistencia. La elección dependerá de la cultura de la Administración y de qué trate el curso de formación. Los manuales y la documentación general suelen estar sólo en inglés y esto podría causar problemas en algunos empleados. La traducción al idioma local podría considerarse como gastos de migración pero entonces aparece el problema de la traducción continuada de las actualizaciones.

La interfaz de usuario de OSS, en concreto, *Gnome* y *KDE*, permite elegir los idiomas pero la traducción puede no ser completa en algunos puntos del menú y las pantallas de ayuda son siempre en inglés. *Gnome* en particular tiene buenas facilidades de acceso para las personas con discapacidad visual. Y además no todas las aplicaciones tendrán pleno soporte de localización. Aunque todo esto está cambiando con rapidez y la estructura que permita el uso de un idioma que no sea el inglés está ahí por si la Administración quiere utilizarla.

Hay ciertas reacciones típicas a los cambios en las prácticas laborales que habrá que afrontar:

- **Miedo a lo desconocido.**

El uso del OSS será completamente nuevo para la mayoría de los usuarios y el personal de sistemas. El miedo a lo desconocido hará que las personas se resistan al OSS porque es nuevo para ellas.

Habrán usuarios que son más curiosos por naturaleza, que pueden sentirse felices de conocer cosas nuevas y son ellos las que deberían probar el OSS en primer lugar. Hasta ahora la experiencia indica que una vez que la gente vence sus reservas encuentra que el OSS no es muy diferente en su uso en comparación con el software propietario y está encantada de usarlo. Por ello es probable que este grupo inicial de usuarios se pase al OSS con entusiasmo. En cualquier caso, esta gente sería también la que proporcione los comentarios y sugerencias más útiles.

El primer grupo de usuarios podría utilizarse en pruebas piloto y una vez que tengan cierta experiencia ya pueden convencer y enseñar a sus colegas. En cualquier caso, ya en la segunda fase, los usuarios que pudieran ser más reservados necesitarán disponer de mayores facilidades de apoyo en forma de ventanillas de atención, intranets y usuarios locales con experiencia.

El mismo proceso se puede usar con el personal de sistemas pero el esfuerzo de formación podría ser importante si el entorno propietario existente no es como UNIX.

El personal de sistemas en particular necesita desterrar sus temores desde el comienzo. Serán un punto focal para todos los problemas que indefectiblemente van a aparecer y si no creen en el proyecto no podrán animar a los usuarios de manera positiva.

- **El temor de que el CV pierda importancia.**

Tanto el personal de sistemas como los usuarios pueden pensar que no usar el software “estándar industrial” perjudicará su capacidad para desarrollar su carrera. Este es un problema delicado que hay que tratar con mucho cuidado. La Administración no querrá verse muy implicada en este enfoque pero hasta que el OSS sea de uso generalizado las Administraciones se pueden encontrar con él con cierta frecuencia.

- **Saber es poder.**

La gente que conoce los sistemas y configuraciones existentes tiene un cierto poder y podrían sentirse bastante reacios a perderlo si el entorno OSS es muy diferente del existente. Y otra vez aparece la necesidad de una gestión cuidadosa ya que esas personas tienen un papel fundamental en el funcionamiento de los sistemas existentes. Quizá sea necesario que estén entre los primeros en recibir formación sobre los nuevos sistemas para que su posición en la entidad se mantenga.

## 7. Una vida más fácil

Hay ciertas circunstancias que pueden hacer que la introducción del OSS sea más fácil.

### 7.1. Introducir nuevas aplicaciones en un entorno familiar

Muchas de las aplicaciones OSS funcionarán con sistemas operativos propietarios y esto nos brinda la oportunidad de introducir estas aplicaciones sin tener que cambiar totalmente el entorno. Por ejemplo *OpenOffice.org*, *Mozilla* y *Apache* funcionará con *Windows* y así puede utilizarse en sustitución de *Office*, *Internet Explorer* e *ISS* respectivamente. Aparte de ser menos rupturista, este enfoque permite que la reacción del usuario pueda ser calibrada a pequeña escala y que los planes para la formación de los usuarios puedan hacerse sobre la base de la experiencia real. Además, problemas como la conversión de formatos de archivos, macros y plantillas se puede facilitar si la antigua aplicación se mantiene disponible durante algún tiempo.

Este enfoque significa que la elección de la aplicación en el entorno final se va a ver limitada a las que trabajan en el actual. Por ejemplo, el navegador final puede ser *Galeon* pero *Mozilla* es el único que funcionará tanto con *Windows* como con GNU/Linux.

### 7.2. Lo fácil primero

Los primeros cambios serán los que no afecten a la comunidad de usuarios. Eso quiere decir que los primeros cambios se harán en el servidor. Estos cambios van a proporcionar la plataforma para la posterior introducción de los cambios en el lado del cliente. Muchos de los cambios relativos al servidor serán compatibles con el entorno actual, con lo que se podrá minimizar el efecto de ruptura.

Por ejemplo, los servidores de nombres DNS, los servidores DHCP y los servidores de bases de datos principales con bases de datos propietarias como *Oracle* podrían ser todos ellos candidatos a ser reemplazados por un OSS equivalente y seguir interactuando con el resto de los sistemas actuales como antes. Más adelante se hablará de esto en detalle.

Hay aplicaciones como *Samba* que no se usarían en un entorno OSS puro, pero que permiten la coexistencia de los antiguos sistemas propietarios y el OSS. El uso temprano de éstas puede ser muy eficaz en la división de los entornos en partes manejables.

### 7.3. Mirar hacia adelante

Evitar hacer cosas ahora que pueda dificultar la migración en el futuro. Por ejemplo:

1. Insistir en que los desarrollos web hechos tanto internamente como por contratistas produzca un contenido que se pueda visualizar en todos los navegadores actuales de la web, en particular los navegadores de OSS. Esta sería una buena práctica en cualquier caso ya que las Administraciones no deberían requerir software específico para visualizar su contenido. Hay herramientas como *weblint* para comprobar la compatibilidad de las páginas web.
2. No fomentar el uso indiscriminado de macros y *scripts* en documentos y hojas de cálculo; encontrar otros modos de proporcionar la necesaria funcionalidad. Ésta también es una buena práctica ya que de forma habitual los virus se valen de las

macros y los *scripts* para infectar los sistemas. Además, las macros se pueden usar fácilmente para robar datos y corromper documentos: por ejemplo, podrían hacer que el documento diga cosas diferentes dependiendo de quien lo esté visionando y que se imprima otra cosa.

3. Insistir en el uso de formatos de archivos abiertos y estándar, como Proscript y PDF.

Hay cierta discusión sobre si Proscript y PDF son estándares abiertos o no. Es más una discusión sobre definiciones estrictas y en concreto sobre quién controla el estándar. En realidad, estos son los únicos formatos de archivos estándar que tienen un amplio uso en este momento, especificaciones públicamente accesibles y que se se pueden usar sin grandes restricciones.

Se están haciendo intentos para crear formatos de archivos estándar basados en XML y *OpenOffice.org* es un candidato. Sin embargo, sólo porque un archivo esté basado en XML ello no significa que vaya a ser abierto.

En particular, no se deben usar formatos de archivos propietarios para archivos que son sólo para lectura y que el receptor no los va a editar. También en este caso sería una buena práctica pues dichos archivos son una forma corriente de difundir virus. Usar esos formatos propietarios significa que la Administración se verá atrapada por el vendedor durante bastante tiempo. Esos formatos propietarios también pueden incluir grandes cantidades de **metadatos** como, por ejemplo, texto previamente borrado, que si otros pueden visionar sería embarazoso para la Administración. Visualizar estos metadatos no es nada difícil

4. Al escribir documentos en colaboración con otros, usar el formato que sea mínimo común denominador. Por ejemplo, hacer uso del formato *Word 97* en lugar de *Word 2000*. Esto aumentará la posibilidad de que las aplicaciones OSS puedan participar.
5. Utilizar protocolos abiertos estándar. Los protocolos abiertos estándar se definen como los que están libres patentes y cuentan con una implantación de OSS. Hay varios conjuntos nacionales de estándares como E-gif en el Reino Unido, OSOSS en los Países Bajos y SAGA en Alemania. El enfoque y contenido de estas infraestructuras es ligeramente diferente pero en general habría que adherirse a ellas.
6. Desarrollar sistemas basados en por lo menos un modelo de tres niveles (véase el punto 8.1) donde el código de aplicación es independiente de la interfaz humana y de los métodos de acceso a los datos. Por ejemplo, si es posible, tener una interfaz de navegador que se pueda usar en un navegador OSS. Construir aplicaciones de esta forma modular facilitará hacer la migración bit a bit. Esto no sólo reducirá la escala de cualquier fase de migración sino que también reducirá el riesgo de fallo. Las tradicionales aplicaciones monolíticas de cliente son notablemente difíciles de manejar.
7. Insistir en que las nuevas aplicaciones se escriban de manera que se sean portables. Esto incluye el usar lenguajes estandarizados portables como ANSI C, Java , Python y Perl, y usar sólo librerías multiplataforma y juegos de herramientas GUI como *wxWindows* (<http://www.wxwindows.org/>) y el *FOX toolkit* (<http://www..fox-toolkit.org/>). Evitar lenguajes y APIs de arquitecturas específicas. Evitar la construcción de aplicaciones que requieran la presencia de otras aplicaciones propietarias.

8. Apartar a los usuarios de lectores de correo propietarios que usen formatos de buzón propietarios y se comuniquen con servidores que usan protocolos propietarios. La mayoría de las aplicaciones de correo guardarán el correo usando IMAP. Si es posible, hallar el modo de guardar la información del calendario y de la libreta de direcciones en formato abierto.

**3<sup>a</sup> PARTE**  
**Directrices**  
**Técnicas**

## 8. Arquitectura de referencia

### 8.1. Arquitecturas genéricas

Una manera útil de describir una arquitectura informática es hacerlo en términos del llamado modelo de tres niveles (*3-tier*). Este modelo diferencia tres grandes funciones que una aplicación realiza en general cuando una persona la utiliza (por ejemplo, no se trata de un simple servidor ni aplicación por lotes). Podemos verlas en el siguiente diagrama. (En <http://www.corba.ch/e/3tier.html/> se puede encontrar información más detallada).



Las flechas indican flujos de información que está circulando entre las tres partes. Estos flujos deberían usar estándares bien definidos y abiertos. Si se hace así cualquier aplicación sólo tendrá que preocuparse de realizar la lógica empresarial dejando las otras dos funciones a los componentes estándar. Esto tiene la ventaja de que el código de la aplicación puede ser más sencillo y se ejecutará con mayor facilidad en diferentes entornos porque su dependencia del acceso concreto a la máquina es reducida.

Este modelo de tres niveles se ha generalizado a  $n$  niveles, donde los componentes están incluso más perfeccionados y es habitual la utilización de tecnología de objetos o componentes.

Lamentablemente muchas aplicaciones de cliente-servidor sólo han usado en el pasado un modelo de dos niveles donde el código de la aplicación y la interfaz humana suelen estar unidos. Esto significa que realizar la migración de esas aplicaciones suele ser mucho más difícil que con los de tres niveles. Y ello se debe a que la interfaz humana parece requerir un cambio y la aplicación de dos niveles tendrá más bien un código de interfaz humana mezclado con lógica empresarial.

La comunicación entre las tres partes de una aplicación de tres niveles utiliza por lo general protocolos que permiten a cada una de las partes, si es necesario, funcionar en una máquina diferente de las otras dos. A veces las partes también pueden dividirse en las máquinas. La elección de la ubicación de cada una de esas partes origina diferentes arquitecturas genéricas.

Los extremos desde el punto de vista del escritorio, donde por lo menos parte del código de interfaz humana debe estar en funcionamiento, son los siguientes:

#### 1. Cliente ligero

Es cuando el escritorio sólo tiene código de interfaz humana. Normalmente no tiene memoria a largo plazo como los discos duros o los disquetes. El código de aplicación y el acceso a los datos funcionan de forma remota. Un ejemplo podría ser un terminal X, una pantalla verde VT100 y un dispositivo con navegador embebido.

#### 2. Cliente pesado

Es cuando todos los códigos y datos se encuentran en un escritorio sin conectividad en red.

Términos como cliente ligero y pesado indican que un escritorio reside en el espectro entre esos dos extremos.

Una variante de estas arquitecturas es aquella en la que el código de aplicación está almacenado en un servidor y se descarga al escritorio para ser ejecutado cuando se necesita. Este es el modo en que trabajan, por ejemplo, los applets (miniaplicaciones) de Java. Otro método implica guardar el código de aplicación en un servidor y hacer que el escritorio acceda a él como si estuviera guardado localmente. Esto exige el uso de sistema de archivos en red como NFS y también significa que todo el escritorio debe tener la misma arquitectura de procesador.

La elección de arquitectura para cualquier aplicación concreta dependerá de:

1. **El ancho de banda de la red para los servidores y qué va a tener que soportar el ancho de banda.** Si el escritorio no es ultra "grueso" entonces la red tendrá que soportar controles de interfaz humana, datos o el código de la aplicación descargada. En algunas circunstancias el tamaño de esas cargas generadas por un escritorio sencillo o varios añadidos unos a otros puede ser demasiado grande para la capacidad de la red.
2. **La latencia aceptable en el uso de la aplicación.** Cuando un ser humano interactúa con el escritorio pulsando teclas o moviendo el ratón, el tiempo necesario para que la aplicación reaccione y muestre el efecto en la pantalla se conoce como latencia. En algunas aplicaciones como la entrada de datos sencillos, se pueden aceptar latencias altas, pero en el caso de aplicaciones muy interactivas como los dibujos, las latencias deben ser bajas. La latencia dependerá de la capacidad de cualquier segmento de red entre la interfaz humana y la aplicación, y de la capacidad de la máquina que ejecuta el código de la aplicación. Por ello, para conseguir las latencias más bajas, la aplicación debería ejecutarse en la misma máquina que la interfaz humana, y la máquina debería tener potencia suficiente para ejecutar la aplicación adecuadamente.
3. **La política de seguridad in situ.** Si los datos residen en máquinas de escritorio distribuidas por toda la Administración, quiere decir que si alguien roba una máquina o se puede acceder a la misma en un entorno poco seguro entonces los datos se pueden perder, llegar a conocimiento de terceros o de personal no autorizado. Esto podría no ser un problema cuando se trata de datos de los que se tiene copia de seguridad, pero si no es así se podría contravenir la política de seguridad de la Administración en relación a quién puede acceder a los datos. Por otra parte, el que los datos se hayan difundido por una red no cifrada podría causar el mismo problema.
4. **La política de backup in situ.** Si los datos residen en máquinas de escritorio distribuidas por toda la Administración, entonces sería necesario algún mecanismo centralizado de backup o bien la responsabilidad de backup se debería repartir entre mucha gente, probablemente los propios usuarios. Un esquema centralizado de backup sería complejo y exigiría un ancho de banda de red alto y cooperación con los usuarios del escritorio (quienes, por ejemplo, deben acordarse de no desconectar su máquina durante los períodos en que se han programado los backups).
5. **El diseño de la aplicación.** Si la aplicación tiene una interfaz humana entonces necesita ejecutarse en el escritorio o bien en un servidor con el código de la interfaz humana dividido entre el servidor y el escritorio. El terminal DEC VT100 o IBM 3270 tiene todo el código de visualización en el escritorio, igual que en todo



terminal basado en el navegador. Citrix, Windows Terminal Server y el Sistema Windows X dividen el código de visualización entre el servidor y el cliente.

6. La capacidad **de la máquina escritorio para ejecutar el código**. Cuanto más tenga que hacer la máquina escritorio, más potente (y por ello más cara) ha de ser.
7. **La capacidad del escritorio para guardar los datos**. Algunas aplicaciones necesitan tener acceso a enormes almacenes de datos que sólo pueden alojarse en servidores especializados.
8. **El rendimiento de los servidores disponibles**. Si la aplicación se ejecuta en un servidor en vez de en un escritorio, entonces el servidor ha de tener la suficiente potencia para ejecutar todas las instancias necesarias de la aplicación cuando el número máximo de escritorios están en uso. Esto podría implicar que los servidores deben ser lo suficientemente específicos para manejar las peores condiciones. Además, el número de escritorios que pueden recibir soporte de un número dado de servidores tiende a oscilar bastante. Esto significa que el añadir unos cuantos escritorios extra implicaría la compra de un servidor extra grande.
9. **El coste total de implantación**. Como en cualquier problema técnico, no hay ninguna solución aplicable a todas las situaciones y un escritorio físico concreto puede funcionar de una manera con una aplicación y de otra con una diferente.

## **8.2. Arquitectura de Referencia**

La Arquitectura Referencia utilizada en estas directrices se ha elegido para ser determinante en la mayoría de las situaciones. Se puede extender para ser más ligera o pesada según aplicaciones concretas si fuera necesario.

En realidad, la arquitectura usada por una Administración suele ser una combinación de varias arquitecturas, cada una de las cuales se ha elegido para aplicaciones concretas.

La Arquitectura de Referencia se puede caracterizar como un "escritorio sin estados" en tanto en cuanto:

1. Todas las aplicaciones se ejecutan en el escritorio siempre que es posible y se almacenan en el escritorio.
2. Los datos no persistentes se guardan en el escritorio.
3. Todas las autenticaciones y autorizaciones son controladas por servidores centrales.
4. La gestión de los sistemas está centralizada.
5. El objetivo es que los escritorios sean "pincha y listo" y no necesitan respaldo local.

Las aplicaciones funcionan localmente para suavizar cualquier problema de latencia al ejecutarlos centralmente y la Arquitectura de Referencia da por sentado que hay suficiente ancho de banda para albergar los datos centralmente. Además, da por sentado que todos los escritorios serán básicamente idénticos, permitiendo que cualquiera se conecte a cualquier máquina que esté autorizado a usar. Debe haber instalado un fuerte régimen de gestión de sistemas para mantener las instalaciones de software de los escritorios al unísono.

La Arquitectura de Referencia tiene una configuración y gestión centralizadas lo que simplificará la administración de los sistemas, concentrando todos los datos importantes en servidores centrales para conseguir una gestión y backup fáciles, y haciendo que las máquinas de clientes individuales estén disponibles y reduciendo así el impacto del fallo de tal máquina.

Mantener los datos localmente significa que hay una identificación de máquina con usuario. Esto causa problemas cuando el usuario cambia de localidad o deja la entidad. También hace que la localidad del escritorio venga determinada por la del usuario, por lo que conceptos como *hot-desking* resultan difíciles. Tener todo los datos centralmente elimina esas dificultades y hace mucho más flexible el uso del escritorio. También permite que el tamaño de la memoria local del escritorio se mantenga en mínimos.

Hacer del escritorio un dispositivo tipo "pinche y listo" facilita la instalación y con ello reduce el coste de mantener el apoyo que se le presta.

Muchas Administraciones ya estarán usando una variante en ese tipo de arquitectura debido a todas las razones dadas y por ello la Arquitectura de Referencia se ve como una buena elección.

La Arquitectura de Referencia no es apropiada para *laptops* o para escritorios que no están permanentemente conectados a la red de la Administración. Esos dispositivos deberían ser muy pesados o tener algún tipo de sistema de archivos repartidos que respalden el funcionamiento desconectado. Existen sistemas de archivos repartidos OSS como *CODA*, *OpenAFS* e *InterMezzo*-, pero **netproject** no los ha testado.

## 9. Grupos funcionales

El modelo de referencia se basa en los grupos funcionales, definiendo los tipos típicos de actividad informática no especializada en una Administración. Esto quiere decir que actividades como la Gestión de Proyectos o los sistemas de Información Geográfica no se contemplan. Las actividades no contempladas deberían ser, en general, las usadas sólo por una pequeña parte de la población de usuarios.

Los grupos funcionales se dividen en grupos Principales y grupos Secundarios. Los grupos Principales representan funcionalmente lo que se ha descrito en términos del proceso de negocio. Los grupos Secundarios prestan servicios de apoyo a los grupos Principales y por ello no deberían ponerse en marcha de manera independiente.

### 9.1. Los grupos principales

#### 9.1.1. Ofimática

Consiste en la creación, modificación e impresión de archivos que contienen datos de negocio formateados estándar, como cartas e informes. También la creación, modificación e impresión de hojas de cálculo y presentaciones. Y debe haber utilidades para gestionar esos archivos. Los formatos de facto de los archivos Microsoft **\*.doc**, **\*.xls** y **\*.ppt** se deben poder leer y escribir con gran precisión. Además, formatos abiertos como PDF se deben poder leer y escribir. Los idiomas locales europeos y configuraciones como la moneda y el alfabeto deben contar con el debido apoyo.

#### 9.1.2. Correo

Consiste en la creación, recepción y visualización de correo electrónico, incluido el apoyo para lograr un correo seguro como S/MIME.

#### 9.1.3. Calendarios y herramientas colaborativas de trabajo en grupo

Consiste en la creación y gestión de calendarios de grupos y libretas de direcciones. Los calendarios deberían permitir concertar reuniones y reservar alojamiento. Las libretas de direcciones se integrarían con los otros grupos funcionales.

#### 9.1.4. Acceso a la web y servicios

Consiste en la capacidad de acceder a los protocolos de servicio de Internet y visualizar los resultados. Esto se hace normalmente con un navegador. Además de la capacidad de crear contenido y hacerlo accesible tanto interna como externamente.

#### 9.1.5. Gestión de documentos

Consiste en el almacenaje central de documentos con mecanismos de recuperación eficaces.

#### 9.1.6. Bases de datos

La manipulación de datos estructurados en base de datos tanto centrales como personales.

### 9.2. Los grupos secundarios

Estos grupos suelen estar definidos por servicios técnicos y normalmente no se implementarían por su cuenta. Entre otros, se incluyen los siguientes:

- sistemas operativos
- servidores de archivos

- gestión de usuarios, autenticación y autorización
- detección de virus y spam
- copias de seguridad y recuperación
- gestión de la impresión

La lista completa se define en el Capítulo 12, más adelante.

### **9.3. Consideraciones generales**

Las administraciones tienen requisitos concretos sobre y en lo que respecta a las necesidades de un entorno de oficina estándar. Algunos de ellos se los impone la legislación local, nacional o europea. En particular:

- Tienen que ser capaces de aceptar archivos del público en formatos de uso corriente. En realidad esto significa aceptar por lo menos todos los formatos Microsoft, pero también podrían tener que aceptar, por ejemplo, formatos *WordPerfect* y *Lotus Notes*.
- Algunas aplicaciones intercambian información entre el público y la Administración, lo que hay que hacer de forma segura con notificación de la entrega.

## 10. El modelo de referencia - Resumen

La amplia gama de alternativas OSS disponible implica que para muchas funciones se dispone de distintas aplicaciones. La elección de la aplicación que se va a usar no siempre está clara y a veces se verá determinada por las preferencias de la persona que toma la decisión.

Por ello, el modelo de referencia empleado en estas directrices debe tratarse como un ejemplo de un sistema del que se sabe que funciona más que como una recomendación sobre el sistema que debería usarse en todas las circunstancias.

En las directrices se habla de los temas que los que toman las decisiones deberían tener en cuenta, y bien puede ser que lleguen a conclusiones diferentes pero igual de válidas. En cualquier caso, las condiciones locales del Administrador pueden determinar que se elija un modelo diferente.

De las posibles opciones de cada grupo se habla más en detalle en el Capítulo 11, en el caso de los grupos principales, y en el Capítulo 12 en el caso de los grupos secundarios.

Hay algunos sitios web de referencia que resultan útiles y disponen de listas de aplicaciones OSS, e indican lo que hay disponible y cuáles de ellas son susceptibles de ser utilizadas para sustituir a ciertas aplicaciones propietarias.

<http://linuxshop.ru/linuxbegin/win-lin-soft-en/> es un ejemplo.

<http://www.osafoundation.org/fesktop-ñlinux-overview.pdf> contiene detalles de muchas de las aplicaciones de las que se habla más adelante y constituye otro lugar en el que los Administradores pueden encontrar mayor información.

Uno de los puntos fuertes del OSS es que es modular y que se puede integrar de formas diversas, permitiendo así que los sistemas se adapten para satisfacer necesidades concretas. Este carácter modular es posible porque OSS se ajusta a las interfaces disponibles abiertas al público. Pero esta flexibilidad, lamentablemente, a veces puede ser su punto débil pues, ya que los Administradores pueden verse intimidados por las opciones disponibles. Hay muchas entidades que pueden prestar apoyo y ayuda, de igual forma que sucede con el software propietario.

Cabe observar que no todos los grupos funcionales disponen de opciones de referencia, bien por la falta de estudios de casos pertinentes bien porque **netproject** no pudo evaluar bien los productos dentro del ámbito de su estudio.

Las Opciones de Referencia detalladas para el escritorio figuran en el Anexo D, y las del servidor Anexo E. Además, **netproject** ha demostrado un modo de instalar los escritorios que hace que esta tarea sea muy sencilla. El correspondiente código se da en el Anexo F.

### 10.1. El escritorio

El sistema operativo es el GNU/Linux de la distribución *Red Hat 8.0*. La interfaz de usuario se basa en Gnome, que forma parte de la distribución Red Hat, pero vale la pena considerar el escritorio Ximian XD2, basado en Gnome. Red Hat, en su distribución, ha tratado de fusionar las interfaces de usuario KDE y Gnome.

Los sistemas de archivos que contienen binarios (como **/usr**) se montan como de “sólo lectura” para evitar que los usuarios alteren sus contenidos, y los restantes se montan “noexec” para impedir que se ejecute el código desde ellos. Para que esto sea posible, la

interfaz de usuario no debería permitir que los usuarios operen programas si no es a través de interfaces predefinidas. Esto significa que se ha de quitar el acceso de línea de comandos o la capacidad para crear o cambiar puntos o iconos del menú. La máquina no debería tener unidad para discos flexibles ni CDs para así restringir (pero no impedir del todo) que otros sistemas de archivos se anexas localmente.

Los sistemas de archivos que contienen datos volátiles basados en el usuario se montan desde un servidor NFS central. La autenticación del usuario se hace con una base de datos LDAP central.

Los servidores DNS centrales hacen resolución de nombres y direcciones con protocolo Internet y un servidor DHCP proporciona los detalles de la configuración en red del escritorio en tiempo de arranque.

Las principales funciones del escritorio se dan como sigue:

#### **10.1.1. Ofimática**

Se elige *OpenOffice.org* porque:

- si la Administración está “emigrando” de un entorno *Windows*, *OpenOffice.org* se puede ejecutar en *Windows*, lo que ofrece a los usuarios un contacto inicial con el nuevo software en un entorno familiar.
- Tiene la mejor integración de formatos de archivo de Microsoft.
- Se está convirtiendo en la alternativa de facto a *Office*.

#### **10.1.2. Correo**

Se elige *Evolution* porque tiene una interfaz de usuario muy similar a *Outlook* y así será fácil de aprender para mucha gente. También tiene algunas características muy útiles como las Carpetas Virtuales. Sin embargo, *Evolution* no se conecta a la versión 5.5. de *Explorer* (aunque parece que hay planeada una) y, si la Administración está usando esta versión, se necesitará una solución de programa de grupos OSS basado en la web en este momento (a menos que se elija una solución propietaria). Por otra parte, *Kmail* presta apoyo a S/MIME mientras *Evolution* no lo hace (aunque está pensado que lo haga) y *Kmail* ha sido creada recientemente para funcionar como un cliente frente al servidor de programa de grupos *Kgroupware*. La elección se ha sopesado muy bien y depende de las necesidades inmediatas y la instalación existente.

#### **10.1.3. Calendarios y herramientas colaborativas de trabajo en grupo**

Se elige *Evolution* para la gestión de contactos y calendarios personales. Es difícil todavía satisfacer con OSS las herramientas colaborativas de trabajo en grupo, y sólo hay por el momento soluciones realmente disponibles basadas en la web, aunque recientemente el proyecto *Kgroupware* ha elaborado una solución utilizando *Kmail* como cliente. Así pues, para una solución verdaderamente OSS se usaría un navegador para el acceso a las herramientas colaborativas de trabajo en grupo.

#### **10.1.4. Acceso a la Web**

Se elige *Galeon* porque es un navegador rápido de función única que tiene una buena interfaz de usuario. *Mozilla* también sería una opción si la Administración está usando escritorios *Windows* en estos momentos, y se necesita el nuevo navegador para funcionar en el entorno existente y que los usuarios tengan el contacto inicial con el nuevo software en un entorno familiar.

### **10.1.5. Gestión de documentos**

Se debería elegir un sistema de gestión de contenidos basado en la web como el *Aswad*. Si bien parece que el proyecto *Aswad* está parado y hay que buscar otra opción.

### **10.1.6. Bases de datos**

Las bases de datos personales se basan en *MySQL* o en un producto de tipo programa de grupos en web como el *phpGroupWare*.

## **10.2. Los servidores**

El sistema operativo es el GNU/Linux de la distribución Red Hat 8.0. Esta elección sería diferente en el caso de máquinas seguras como los cortafuegos, con los que podría usarse OpenBSD en combinación con GNU/Linux.

Las principales funciones de servidor se prestan mediante:

### **10.2.1. Correo**

El MTA (Agente de Transporte de Correo) es el *Exim*, porque se trata de un producto totalmente desarrollado comparable al *Sendmail* en cuanto a su alcance, pero de mantenimiento más sencillo. También entiende que las opciones de *Sendmail* entonces se pueden ejecutar como sustitución compatible de *Sendmail*. *Postfix* sería igualmente una alternativa aceptable.

### **10.2.2. Calendarios y herramientas colaborativas de trabajo en grupo**

*PhpHroupWare* o *Horde* serían muy buenas soluciones basadas en la web. Aún no se ha evaluado el nuevo *Kgroupware*.

### **10.2.3. Acceso a la Web**

Se elige *Apache* porque es el líder del mercado con una amplia gama de herramientas y respaldo. Para tareas específicas se podrían usar otros servidores como, por ejemplo, *Zope* (véase el punto 11.4.2. más adelante) para la gestión de contenidos.

### **10.2.4. Gestión de documentos**

Ahora el *Aswad* (véase el punto 11.4.3.) parece estar parado y no hay solución de referencia. En el punto 11.5 se habla de la existencia de cierta opción disponible.

### **10.2.5. Bases de datos**

En cuanto a las bases de datos grandes, que suelen ser de sólo lectura, está *MySQL*; para otros tipos de bases de datos, *PostgreSQL*.

## 11. Aplicaciones – Grupos principales

### 11.1. Ofimática

El estándar actual es, de hecho, *Office* que incluye *Word*, *Excel*, *PowerPoint* y *Outlook* con sus correspondientes formatos de archivo: **\*.doc**, **\*.xls** y **\*.ppt**. Estos formatos no son abiertos y cambian de una versión de *Office* a otra. Incluso los propios productos de Microsoft no pueden garantizar la capacidad para leer y escribir un archivo con una precisión del 100% a menos que los archivos se hubieran creado con la misma versión de su producto.

Ahora las aplicaciones OSS ya pueden leer estos formatos con bastante precisión y los problemas encontrados no son diferentes a los que aparecen al usar diferentes versiones de los propios productos de Microsoft. Cuanto más antiguo sea el formato, mejor actúan las aplicaciones OSS. Las aplicaciones OSS tienden a ser mejores en la lectura de archivos que en su escritura en esos formatos. Por ello se pueden usar las aplicaciones OSS, en general, con confianza. Ximian, por ejemplo, en su último producto para escritorio, ha establecido en su versión de *OpenOffice.org* que los formatos de archivo por defecto sean los de Microsoft.

La excepción aparece cuando se necesita algún tipo de trabajo de colaboración y al menos una de las partes insiste en usar un formato propietario. La lectura, alteración y reescritura de archivos en estos formatos pueden introducir anomalías que no se darían con el uso de una sola aplicación propietaria. Pero se debe tener siempre presente que este tipo de corrupción también puede ocurrir si se usan diferentes versiones de software propietario.

En el caso de archivos que son sólo para lectura y ni están actualizados, se debería usar el formato PDF, tal y como se indica en el punto 7.3 anterior.

Algunos pueden creer que la interfaz de usuario debería ser lo más parecida posible al software de Microsoft para minimizar los costes de nueva formación.

Las plantillas y las macros de *Visual Basic* (VB) son corrientes en muchas Administraciones. Se encuentran igualmente en un formato propietario cerrado y será necesario rescribirlas.

Hay tres paquetes de ofimática OSS, *OpenOffice.org*, *KOffice* y *Gnome Office*, que debieran ser tenidos en cuenta.

Ya se dispone en la web de un estudio piloto comparando la capacidad de los diversos paquetes de ofimática OSS para manejar los archivos de *Microsoft Office*:

<http://www.acmqueue.com/modules.php?name=Content&showpage&p=5>

#### 11.1.1. OpenOffice.org y StarOffice

*OpenOffice.org* es un paquete de ofimática OSS basado en *StarOffice* que creó una empresa alemana llamada *StarDivision*. Sun Microsystems compró a *StarDivision* y aportó el código a la comunidad OSS. Sigue comercializando una versión de *OpenOffice.org*, aún denominada *StarOffice*, que se vende a un precio mucho más bajo que otros paquetes propietarios comparables.

*StarOffice* y *OpenOffice.org* son básicamente idénticas, excepto en lo siguiente:

- Sun Microsystems brinda respaldo comercial a *StarOffice*.



- *StarOffice* tiene un sistema de bases de datos incorporado (Adabas) .
- *StarOffice* tiene algunos filtros extras para migración a/de otros paquetes de ofimática (aunque el filtro *Wordperfect* no está disponible en GNU/Linux por cuestiones de licencia).
- *StarOffice* tiene algunos tipos de letras propietarios.
- *StarOffice* sólo está disponible en una serie de idiomas algo limitada (alemán, español, francés, holandés, inglés, italiano y sueco).
- *OpenOffice.org* se actualiza con más frecuencia que *StarOffice* .

Las dos aplicaciones son similares a *Office* pero no llevan cliente de correo electrónico. Las dos manipulan la mayoría de archivos *Office* e incluyen *Office XP*, aunque la compatibilidad va empeorando con las versiones de *Office* posteriores a *Office 97*. No manipulan archivos protegidos con contraseña (excepto los de protección a nivel de hoja en las hojas de cálculo) y tienen algunos problemas con los objetos gráficos vinculados a OLE.

Tienen su propia implantación de Basic y no pueden manipular macros de *Visual Basic*, que deben traducirse manualmente. Aunque la traducción es un coste de migración, la falta de apoyo a las macros evita que los virus de macro se transmitan. Sun Microsystems está estableciendo enlaces con empresas para traducir las macros y plantillas de Microsoft a una forma compatible con *StarOffice*.

También ofrecen una interfaz *Java*, pero actualmente sólo reconocen la Microsystems JDK. Sun Microsystems ha anunciado un proyecto para elaborar un traductor de *Visual Basic for Applications (VBA)* a *Java*.

La versión que se puede descargar de *OpenOffice.org* no contiene más que unos cuantos diccionarios, pero hay diccionarios disponibles en la mayoría de los idiomas europeos. Ya existe una versión preinstalada de 26 idiomas diferentes.

Aunque en la actualidad *OpenOffice.org* no ofrece todavía un paquete de bases de datos, sí tiene interfaces ODBC y JDBC para muchos sistemas corrientes de bases de datos como, entre otros, los populares OSS. Tampoco hay filtros de conversión para *Wordperfect*, pero se ha previsto que aparezcan en el futuro.

Los dos funcionan con diversos sistemas operativos como, por ejemplo *Windows* y *GNU/Linux*.

Los revisores están empezando a clasificar a *OpenOffice.org* de cara a la funcionalidad y la estabilidad – véase:

<http://www.raycomm.com/techwhirl/magazine/technical/openofficewriter.html>

El siguiente sitio proporciona un buen manual de *OpenOffice.org*:

<http://www.taming-openoffice-org.com/>

### 11.1.2. Koffice

Este es el componente de ofimática del escritorio KDE. Es un paquete integrado con herramientas de proceso de textos, hoja de cálculo, gráficos, presentación, ilustración, creación de informes y diagramas de flujo con un escritorio opcional denominado Workspace.

Los filtros de archivos Microsoft no son tan buenos como los proporcionados por *OpenOffice.org*. No tienen lenguaje de macros, pero sí *scripting* (archivos de comandos).

### 11.1.3. Gnome Office

Consiste en un conjunto de programas escritos para las normas Gnome y por ello pueden integrarse unos con otros, tener una interfaz de usuario similar y también podrían integrarse.

Ahora *OpenOffice.org* se considera parte de *GnomeOffice*, aun cuando no se adhiere a las normas Gnome. Ximian, en particular, trabaja para hacer que *OpenOffice.org* sea más compatible con Gnome y ha incluido su propia versión en su último producto de escritorio, XD2. (Véase <http://www.gnome.org/gnome-office/> para mayor información).

*Gnome Office* tiene muchos componentes como, por ejemplo, *AbiWord* (proceso de textos), *Gnumeric* (hoja de cálculo), *Sodipodi* y *Sketch* (dibujo de gráficos de vectores), *Gimp* (edición de imágenes), *Eye of Gnome* (visualización de imágenes), *Dia* (gráficos de vectores, similares a *Visio*) y *Agnubis* (gráficos de presentaciones).

Estos componentes tienen distintos grados de capacidad de utilización; *AbiWord*, por ejemplo, es muy bueno en el proceso de textos básico pero no tanto con las tablas, y *Agnubis* es muy limitado, mientras *Gnumeric* es una hoja de cálculo de gran capacidad. Se sigue trabajando activamente en el perfeccionamiento de casi todos los componentes.

*Gnumeric*, en especial, tiene el objetivo de elaborar una hoja de cálculo que pueda hacer todo lo que hace *Excel*, y mucho más. La última versión de tests beta de *Gnumeric* (1.1.20 cuando se escribe este informe) da apoyo a todas las funciones de hojas de trabajo de la versión estadounidense de *Excel*, y se espera que en septiembre de 2003 salga una versión de producción que incluya este nivel de funcionalidad. Los creadores proceden del ámbito financiero y han incluido una serie de características que hacen a *Gnumeric* especialmente útil para aplicaciones financieras. Es en esta área donde creen que *Gnumeric* supera a *Excel*.

*Gnumeric* utiliza **\*.xls** como formato de archivo nativo, mientras que *OpenOffice.org* convierte las hojas de cálculo a un formato de tipo XML.

La gama de productos disponible es interesante y, junto con *OpenOffice.org*, proporciona varias soluciones diferentes. Pero si se requiere un paquete integrado, entonces *OpenOffice.org* es la única solución real.

## 11.2. Correo

La del correo es un área compleja con varios componentes lógicos, y tiene profusión de aplicaciones OSS, algunas de las cuales proporcionan una funcionalidad parcialmente coincidente. También está estrechamente ligado a otros temas, como el control de los virus y el spam (correo no deseado, o "correo basura").

La elección de aplicaciones adecuadas es compleja, y en el Anexo C se habla con detalle de estos temas, y se da la definición de los términos usados aquí. 13.3.

### 11.2.1. MTA

Los principales MTA de OSS son *Sendmail*, *Exim*, *Courier- MTA* y *Postfix*.

Hay muchos más pero estos son los más utilizados para uso a gran escala. Tradicionalmente, los sitios de Unix y OSS han utilizado *Sendmail* como MTA. Pero su índice de seguridad ha sido muy bajo y también resulta muy difícil de configurar.

Todos los demás tienen buena reputación y a nivel técnico existen pocas diferencias que nos lleven a optar por uno u otro. No obstante, existe una notable diferencia en cuanto a la documentación disponible, estando *Postfix* mejor documentado a nivel inicial y *Exim* a nivel de expertos.

*Exim* y *Postfix* se incluyen en algunas distribuciones OSS pero pueden no ser las soluciones por defecto.

*Courier-MTA* viene como parte de una familia formada por MTA, MDA (Agente de Entrega de Correo), MAA y un paquete de correo en web (*sqwebmail*). Cada una de estas partes se puede usar por separado, o integrada con el resto de la familia.

*Qmail* es un MTA que se suele confundir con OSS. Aunque se dispone del código fuente, tiene una licencia de *copyright* que sólo permite la distribución de versiones estrictamente idénticas a las distribuidas por su autor; así pues, se trata de software propietario. Su licencia restrictiva hace que a los vendedores de distribuciones para OSS les resulte extremadamente difícil darle apoyo o integrarlo en su otro software, y por ello no lo distribuyen. Sin embargo, tiene un buen índice de seguridad y se usa mucho.

Otro producto interesante es el servidor de correo *James*, de Apache. Pretende ser un MTA completo escrito en Java. De momento le faltan algunas características, pero vale la pena no perderlo de vista de cara al futuro.

La opción recomendada es *Exim*. Y ello porque tiene tanta capacidad como *Sendmail*, a la vez que es fácil de configurar y probablemente más seguro. Los otros dos son quizá menos capaces de manejar grandes volúmenes de mensajes. Pero la elección no es indiscutible y cada uno debería hacer su propia elección a partir de las necesidades locales.

### 11.2.2. Buzón

La mayoría de las Administraciones querrían que los clientes usaran un buzón de correo centralizado en lugar de que descarguen los mensajes a un buzón local en el cliente de escritorio. Por ello recomendamos sinceramente el uso de IMAP.

Hay tres IMAP de OSS muy conocidos: *UW-IMAP* (a veces denominado simplemente IMAP), *Courier-IMAP* y *Cyrus*.

*UW-IMAP* tiene un historial de escasa seguridad y no se recomienda.

De los otros dos, *Courier-IMAP* se considera en general el más fácil de configurar. Tiene un rastro más pequeño y trabaja bien con *Postfix* y *Courier-MTA*. Es la parte MAA de la familia Courier. Requiere maildir como el formato de almacenaje de correo. Se dice que con él es difícil tratar algunos mensajes S/MIME, ya que desplaza los encabezamientos de manera que invalida la firma.

*Cyrus* usa su propio formato de almacenaje de correo, que es similar a maildir y requiere su propio MDA para “poblar” el buzón.

Tanto *Courier-IMAP* como *Cyrus* dan apoyo a **TLS** (un protocolo estándar de autenticación y privacidad).

Hay ciertos MDA, por ejemplo *procmail*, el *maildrop* de Courier que viene como parte de la familia Courier y *Cyrus'deliver*. Los MDA también pueden filtrar

correo según sofisticadas normas que son útiles si el MUA utilizado no tiene dispositivos de filtrado.

La elección de referencia fue **Courier-IMAP** sin MDA. No se necesita un MDA, porque **Exim** puede escribir directamente a estructuras maildir y **Evolution** tiene sus propios y potentes filtros.

### 11.2.3. MUA

Hay muchos MUA de tipo texto o de interfaz gráfico disponibles en el campo de OSS.

En el caso de quienes están habituados a *Outlook* u *Outlook Express* y quieren algo similar, la elección obvia es *Evolution*. *Evolution* no es sólo un cliente de correo sino también un Gestor de Información Personal (PIM). Tiene integración LDAP y por ello puede acceder al nombre y la dirección de las Administraciones siempre que esté en el esquema de *Evolution*. *Ximian* lo está desarrollando con rapidez como su producto enseña. *Ximian* hace un producto llamado *Connector* que permite a *Evolution* conectarse a *Exchange* (pero no en la versión 5.5). *Connector* es propietario y tiene pago por licencia

Lamentablemente, *Evolution* no da pleno apoyo al IMAP en modo desconectado. Sólo copia cierto correo al cliente, no todos. No obstante, tiene una prestación muy buena denominada "carpetas virtuales" que permite al usuario determinar normas para ver su correo de muchas maneras diferentes sin tener que sacar múltiples copias del mismo.

*kmail* y *sylpheed* son alternativas a los MUA. Los dos son muy buenos y se integran con los principales entornos de escritorio OSS. *kmail* se usaría si en el escritorio hay un KDE, y *sylpheed* si el escritorio es Gnome.

*Evolution* da apoyo a GPG pero no a PGP aunque pronto lo hará.

*Mozilla* da apoyo a S/MIME, GPG y PGP como parte del proyecto Ägyptens financiado por el gobierno alemán.

Muchos paquetes de herramientas colaborativas de trabajo en grupo incluyen también clientes compatibles con IMAP y POP3. En general no son tan buenos como *Evolution*, pero pueden bastar si se integran bien en las otras funciones de programa de grupos.

En algunos casos, puede ser mejor trasladar ciertas categorías de usuario de correo a una interfaz de usuario de tipo web. *SquirrelMail* es una especialmente buena que se puede encontrar en <http://www.squirrelmail.org/>

La *Open Systems Applications Foundation* hace un producto llamado *Chandler* que está en sus inicios pero al que vale la pena seguir la pista de cara al futuro. Es un potencial rival de *Evolution*.

Como es posible que la mayoría de los usuarios vayan a necesitar algo que se parezca al Outlook de Microsoft, *Evolution* sería la opción preferida, y de hecho es la opción de recomendada. No obstante, si se necesita un S/MIME de inmediato, entonces se debe usar *Kmail* lo que puede significar usar KDE en lugar de Gnome.

### 11.2.4. Antivirus

Si los sistemas OSS están configurados correctamente, los virus tienen un efecto limitado. Pero está el problema del paso de virus a sitios que tienen otros sistemas operativos, incluyendo, en particular, los productos de Microsoft.

Hay un producto antivirus para OSS, el *ClamAV*, pero presenta algunos problemas. Por ello en estos momentos sólo se recomiendan productos propietarios con pleno apoyo.

El mejor modo de ejecutar esos productos es hacerlo como parte del MTA. Tanto *Postfix* como *Exim* proporcionan los medios para incorporar esos filtros.

Hay una opción de productos propietarios. Tanto *Sophos* como *RAV* y *Vexira* están bien considerados. *Trend* es bueno comprobando si hay virus pero debe ejecutarse con súper acceso de usuarios, lo que plantearía un mayor problema de seguridad si estuviera comprometido.

No se hace recomendación alguna pues **netproject** no ha podido probar totalmente estos productos.

#### **11.2.5. Otras herramientas**

Hay algunas herramientas anti *spam* (correo basura) que evitan que archivos adjuntos ejecutables se descarguen con el correo.

*SpamAssassin* es probablemente la más conocida de las herramientas anti-spam que analizan los mensajes.

*Anomy Sanitizer* es una herramienta configurable capaz de eliminar los archivos adjuntos de los mensajes. Es configurable pero hay que usarlo con cuidado pues la eliminación puede invalidar la firma de un documento firmado.

*MailScanner* es un marco general para el control de los contenidos, e incluye medidas anti-virus y anti-spam. Puede recurrir a uno o más productos anti-virus propietarios que actúan sobre mensajes de correo electrónico y también puede usar *SpamAssassin*, así como aplicar sus propias reglas.

*Fetchmail* recoge el correo electrónico de un buzón de correo remoto y lo guarda o lo pasa a otro MTA. Como entrega el correo en una máquina (p. ej. la transferencia la inicia el receptor) resulta útil cuando, por razones de seguridad, los Administradores no quieren abrir un puerto en su máquina de acceso a Internet para que el correo electrónico sea “empujado” hasta ellos (donde el remitente inicia la transferencia) como sucedería con el modelo SMTP normal.

*OfficeImap* es una herramienta que permite que un buzón de un servidor central esté sincronizado con el buzón de un cliente. Para ello el cliente se conecta regularmente a un servidor que utiliza IMAP. La estructura local es maildir. Y esto puede ser muy útil para permitir que el IMAP en modo desconectado sea emulado si el MUA no le da apoyo pleno.

*Whoson* permite el método de autenticación POP-before-SMTP de los usuarios remotos. Es necesario su uso si los usuarios han de poder enviar correo electrónico a través de servidores de la Administración de forma remota si el SMTP autenticado no está disponible y el MTA de la administración está abierto a conexiones de fuera de su gama de confianza.

#### **11.2.6. Problemas experimentados**

Guardar datos en un servidor LDAP requiere que se elija un esquema. El esquema debe ser compatible con todos los clientes que puedan tener que acceder a los datos. Afortunadamente, algunos paquetes vienen con un esquema que no sólo da apoyo a sus necesidades sino también a las de otros paquetes.

*CourierIMAP* viene con un esquema pero *Exim* no. El esquema de *Courier* da apoyo también a *Exim*. No sabemos si da apoyo a todas las capacidades de *Exim*.

Se descubrieron algunos problemas con el archivo de configuración LDAP para *Courier*. Se le han enviado correcciones al creador pero aún no se han incluido en ninguna versión.

Usar *Courier* con *Whoson* exige hacer algunas modificaciones a *Courier*. Algunas se encontraron en el sitio de *Whoson* pero eran bastante antiguas y había que adaptarlas mucho para trabajar con la versión seleccionada de *Courier*.

### **11.3. Calendarios y herramientas colaborativas de trabajo en grupo**

Los calendarios son un asunto mal definido en OSS. Ello se debe a la falta de normas estándar para la comunicación entre los clientes y el servidor central. Por eso los productos creados hasta ahora usan una entrega tipo web y esto puede que no dé el aspecto visual y operacional al que la gente se ha acostumbrado con *Exchange* y *Outlook*. Este área constituye una gran desventaja en la cartera de productos OSS.

Los productos reseñados en la siguiente tabla serían usados como entrega de tipo web a menos que se establezca otra cosa. Todos forman parte de paquetes de herramientas colaborativas de trabajo en grupo que tienen una amplia variedad de características extra.

La mayoría de productos de herramientas colaborativas de trabajo en grupo se escriben en PHP o Perl, y por ello se pueden hacer a medida. En esos productos se ha incluido alguna integración de facilidades interesante.

*phpGroupWare* (<http://www.phpgroupware.org/>) tiene buena reputación.

## Detalles de los productos de herramientas colaborativas de trabajo en grupo

<i>Producto</i>	<i>E-mail</i>	<i>Calendarios</i>	<i>Gestión de documentos</i>	<i>Chat</i>	<i>Lista de tareas</i>	<i>Gestión de contactos</i>	<i>Base de datos</i>	<i>Hoja de tiempos</i>	<i>Programación</i>	<i>Otras facilidades</i>	<i>Observaciones</i>
<b>NullLogic</b>	Y	Y	Y	[1]	Y		Y				
<b>Twiki</b>		Y	Y	Y			Y				Más un marco que un producto
<b>PhpGroupWare</b>	Y	Y	Y		Y	Y	Y		Y		Difícil encontrar información precisa en el sitio web
<b>Project</b>	Y	A, B	Y		[2]	Y		Y		Gestión de Proyectos, Marcadores, Recordatorio, Sistema de búsqueda, sistema de votación, seguidor de consultas	
<b>Tutos</b>	Y	B	Y			Y		Y			
<b>Twiggi</b>	Y	Y			[2]	Y			Y	Para Hacer, Marcadores	

### NOTAS

- 1 Estándar de Forum IM plus (BBS)
- 2 Breve funcionalidad tipo “nota pegada”
- A Registro de hojas de tiempos; Seguimiento de consultas, una forma de gestión de flujos de trabajo: lista de “Para Hacer”; Proyectos dinámicos
- B Recursos (p. ej. Salas de reuniones, retro-proyectores) y Registro de Eventos (futuros y pasados)

*horde* es un marco para ejecutar otras aplicaciones. Por ejemplo, *Imp*, un servidor de correo de web, *Turba*, un gestor de contactos y *Kronolith*, un calendario. Véase <http://www.horde.org/>.

*NullLogic* aparentemente sólo ofrece interfaz en inglés pero *phProject*, *Tutos*, *Twiggi* y *Twiki* dan apoyo a varios idiomas.

Un producto muy reciente de OSS es el *OpenGroupware* de <http://www.opengroupware.org/>. Es la antigua aplicación propietaria *SKYRIX* que sus propietarios han convertido en OSS. Se pretende que sea un sustituto de *Exchange*. No ha habido tiempo para estudiarlo a fondo pero en principio parece que va a tener gran influencia.

Otro producto reciente es el *Kgroupware* (Véase <http://kolab.kroupware.org/>). Este producto tiene un cliente basado en *Kmail* que vale la pena investigar si se elige KDE como interfaz de usuario o *Kmail* como el MUA para dar apoyo a S/MIME.

### **11.3.1. Calendarios y agendas personales**

Todos los productos pueden mantener calendarios personales y listas de cosas para hacer a menos que se establezca otra cosa.

### **11.3.2. Calendarios de grupo**

*Tutos*, *Twiggi* y *NullLogic* dan apoyo a los calendarios de grupo. *Tutos* se puede controlar en niveles que van desde trabajo individual a grupo de trabajo o grupo de proyectos a cualquiera.

En *NullLogic*, no se puede mantener la privacidad de los calendarios respecto a otros miembros del grupo, pero sí se puede con las tareas.

### **11.3.3. Organización de reuniones**

Muchos de los productos incorporan características para la programación de recursos que se pueden usar para planear reuniones.

*Tutos* permite la asignación automática de personas, así como la notificación por correo electrónico a las que no están en el calendario compartido (como las de otras entidades). Sigue la pista de las aceptaciones y enviará recordatorios por correo electrónico en caso necesario. *phProject* es similar, y también hará notificaciones de texto en SMS.

*NullLogic* da apoyo a las características arriba indicadas, excepto la asignación de alojamiento.

### **11.3.4. Sincronización de PDA**

*phProject* tiene un añadido que se sincroniza con los PDA de tipo PalmOS. A la sincronización PDA también se le da apoyo como parte de *Gnome* y *Evolution*. Se pueden sincronizar los PDA más conocidos.

## **11.4. Servicios de Web**

### **11.4.1. Navegador**

Los principales navegadores OSS son *Mozilla*, *Galeon* y *Konqueror*. Hay otros, como *Lynx* que es sólo de texto, y se suele usar como base para los navegadores destinados a personas con discapacidades, y el *Firebird* de *Mozilla* (antes conocido con el nombre de *Phoenix*), una variante ligera de *Mozilla*. También hay un navegador propietario, *Opera*, con una versión que funciona con



GNU/Linux. *Netscape* se basa en *Mozilla* y funciona en plataformas OSS, pero incluye algún código propietario.

*Mozilla* es el principal proyecto OSS basado en un código liberado por *Netscape*, y es la base de *Netscape* 7. Contiene componentes de noticias y correo, así como una libreta de direcciones y una herramienta que autoriza la web. Mucho del código de *Mozilla* es usado por otros proyectos como, por ejemplo, *Galeon* y *OpenOffice.org*.

*Galeon* es sólo un navegador y está pensado para que sea pequeño y rápido. Se basa en *Gecko*, la máquina de síntesis de imágenes de la que nace el proyecto *Mozilla*, junto con una interfaz de usuario de *Gnome*.

Tanto *Galeon* como *Mozilla* soportan todas los estándares abiertos web de Internet y pueden ejecutar código Java y Javascript bien escrito.

Algunos contenidos requieren un accesorio que sólo está disponible para *Windows*, como *Schockwave Director*. El producto propietario *CrossOver Plugin* de *CodeWeavers* permite que *plugins* que funcionan con *Windows* se ejecuten con GNU/Linux.

*Konqueror* es el navegador de web escrito para el escritorio KDE y se usa también como un gestor de archivos de arrastrar y soltar. Se basa en la máquina de síntesis de imágenes de KHTML, con el *Gecko* de Mozilla como opción, y con una interfaz de usuario KDE.

#### 11.4.2. Servidores Web

El servidor web OSS más conocido es *Apache*, que, según el estudio de *Netcraft* (<http://www.netcraft.com/>) tiene más del 60% del mercado y este porcentaje va en aumento. Una combinación de productos que cada vez tiene más aceptación se agrupa con el nombre de LAMP: *Linux*, *Apache*, *MySQL* y *PHP*. Esto proporciona un marco para los sitios web que acceden a las bases de datos en SQL a través del lenguaje PHP. Todos los componentes son OSS.

El proyecto *Apache* contiene varios proyectos secundarios, uno de los cuales se denomina *Jakarta* y cubre el uso como servidor de Java. El propio *Jakarta* está formado por proyectos secundarios, dos de los cuales son *Tomcat* y *Slide*. *Tomcat* proporciona un producto para los servlets de Java según las normas JSP, y la capacidad para usar tecnologías como la *Websphere* de IBM. *Slide* es una puesta en práctica tipo Java de WebDAV que permite la gestión de contenidos. Para más información, consulte <http://www.apache.org/>.

Otros servidores que vale la pena tener en cuenta son *Zope* y *Tux*.

*Zope* (<http://www.zope.org/>) está pensado para proporcionar apoyo dinámico a los contenidos de web y se basa en un modelo orientado al objeto. Constituye un paquete interesante, ya que combina un sistema de gestión de contenidos con un servidor de web y un sistema de plantillas en un solo paquete. *Zope* también da apoyo a añadidos modulares (llamados productos) y se basa en el lenguaje Python orientado al objeto. Es corriente encontrar a *Zope* desplegado "detrás de" *Apache* en una configuración multiservidor, donde *Apache* sirve el contenido estático y actúa como acelerador tipo cache para las partes del sitio gestionadas por *Zope*.

*Tux* es una creación de Red Hat que ahora recibe el nombre de *The Red Hat Content Accelerator*. Utiliza un kernel especial y se supone que proporciona una rápida respuesta a las páginas estáticas.

*Roxen* es otra combinación de servidor de web y sistema de gestión de contenidos, aunque para una plena funcionalidad de la gestión es necesario comprar añadidos propietarios.

De estos, *Apache* es con mucho el más popular. Suele ejecutar el 63% de los sitios web públicos en todo el mundo y su cuota de mercado está aumentando constantemente a costa de *IIS*, por lo que hay mucha experiencia para saber cuándo decidir una migración. *Apache* es un servidor modular, con un motor de protocolo central y una gran selección de módulos para fines específicos.

#### 11.4.3. Portal / Contenido

*Zope*, junto con otros componentes OSS, forman parte de un proyecto financiado por la UE y denominado *ASWAD* (<http://www.aswad-project.org/>) que está pensado para la gestión de contenidos. **netproject** esperaba poder juzgar este proyecto, pero tuvo dificultades para conseguir información actualizada sobre el mismo. *Plone* es un interesante proyecto basado en *Zope* (<http://www.plone.org/>).

*JBoss* (<http://www.jboss.org/>) es un servidor de aplicaciones basado en Java. Tiene una buena reputación y está elaborado activamente.

Ahora hay muchos productos para la gestión de contenidos OSS, como podrán ver si visitan <http://www.oscom.orgg/matrix/index.html> **netproject** no encontró información detallada en los estudios de casos publicados y no pudo investigarlos tan a fondo como era preciso para elegir un candidato para el Modelo de Referencia.

### 11.5. Gestión de documentos

#### 11.5.1. Registro y recuperación

La Gestión de Documentos puede, y quizá debería, verse como una forma de gestión de contenidos y de flujo de trabajo. Este es el tipo de cuestión que *ASWAD* (véase el punto 11.4.3 anterior) pretende cubrir. Por ello **netproject** recomendaría la adopción de una solución de entre las soluciones para la gestión de contenidos disponibles. En concreto, las que usan WebDAV podrían brindar las soluciones más útiles.

Un estándar alemán llamado *DOMEA* (*Disposition and Archiving of Electronic Records* - Disposición y archivado de registros electrónicos), que no se usa mucho fuera de Alemania, ha sido adoptado por IBM junto con *SAP*. La mayoría de los documentos relativos a *DOMEA* (en particular los encontrados mediante una búsqueda en Google) están escritos en alemán. Hay una empresa llamada *FabSoft* que da apoyo a *DOMEA* sobre GNU/Linux en macroordenadores IBM. Parece que no hay ningún producto OSS que dé apoyo a *DOMEA*.

Algunos de los productos groupware dan apoyo a la gestión de documentos:

*Tutos* incluye un sistema para la gestión de documentos que también tiene gestión de versiones.

*NullLogic* incluye una sencilla capacidad de almacenaje, indexación y descarga de archivos. No parece ofrecer un sistema de gestión de cambios. Tiene un mecanismo generalizado de consultas que podría instalarse para ofrecer una indexación sensible.

### 11.5.2. Trabajo en colaboración

Esta función se puede poner en marcha *ad-hoc* mediante el simple intercambio de documentos entre personas. El intercambio se puede hacer por archivo adjunto de correo electrónico o mediante mecanismos como los usados por CIRCA.

La colaboración exige a las partes que se pongan de acuerdo en el formato del documento, y actualmente mucha gente usa **\*.doc** de Microsoft de manera predeterminada. Esta predeterminación significa que las partes han de confiar las unas en las otras, porque esos formatos pueden ser eficaces vehículos para difundir virus. Además, como un estándar, **\*.doc** no resulta ideal porque está cambiando constantemente; el formato tal y como lo usa *Office 2000* no es idéntico al utilizado por *Office 97*. Esto significa que las partes también tienen que ponerse de acuerdo sobre qué versión de software usar.

Hay cierta presión para que se adopten formatos de documentos basado en estándares, en particular los basados en XML. *OpenOffice.org* proporciona un estándar de documentos abierto de tipo XML que se podría usar como base de colaboración.

Un planteamiento más estructurado sería adoptar una solución para flujo de trabajo/gestión de contenidos tal y como se describe anteriormente.

El producto de groupware *Tutos* permite que los documentos estén sujetos al control de una sola persona o de todas dentro de un grupo concreto. *NullLogic* y *Twiki* también han perfeccionado los controles.

En cuanto a la gestión de contenidos, no hay candidato para el Modelo de Referencia.

## 11.6. Bases de datos

### 11.6.1. Bases de datos generales según aplicaciones

Los sistemas de bases de datos OSS disponibles incluyen *MySQL*, *PostgreSQL* y *Firebird*, que tienen una aplicabilidad y unas características muy diferentes.

*MySQL* es una base de datos en SQL ligera especial para servidores web y aplicaciones similares. Es adecuada en situaciones en las que predomina la lectura sobre la escritura. No da apoyo a procedimientos de bases de datos ni consultas secundarias complejas. Está previsto el apoyo para los procedimientos de bases de datos en la versión 5.

*PostgreSQL* es un DBMS totalmente compatible con *Oracle* y *DB2*, pero sin las características más avanzadas necesarias para manejar grandes volúmenes de datos.

*Firebird* es una versión de la base de datos *Interbase* de Borland que salió con licencia OSS. Gran parte del código lo comparte con *Interbase* y como tal debe considerarse maduro. Existe el proyecto de añadir capacidad de base de datos a *OpenOffice.org* usando *Firebird* pero está en una etapa inicial.

### 11.6.2. Bases de datos personales de entorno local o central

Las bases de datos personales no están bien soportadas por OSS. No hay un equivalente directo a *Access* ni está en preparación.

Varios de los paquetes de groupware ofrecen una cierta capacidad en esta área usando una variedad de bases de datos SQL en OSS como motor. En algunos casos (como *NullLogic*) los usuarios corrientes sólo pueden usar consultas predefinidas. Algunas ofrecen la posibilidad de definir formularios que se pueden usar para guardar datos y acceder a ellos.

### **11.6.3. Conectividad de bases de datos**

La mayoría de los productos DBMS dan apoyo a los API directos con enlaces de lenguaje C. Algunos también dan apoyo a C++ originariamente.

Todos ofrecen conectividad ODBC y JDBC. Algunos también ofrecen conectividad NET.

Hay un producto llamado *UnixODBC* que proporciona a los programas Unix y GNU/Linux una conectividad similar a la ODBC.

### **11.6.4. Rendimiento**

El rendimiento de las bases de datos depende en gran medida del tamaño de las tablas implicadas y la complejidad de las consultas.

Ninguno de las alternativas OSS podrían hacer frente a las demandas de las que sí pueden ocuparse bases de datos como *Oracle*. Esto es especialmente así en aplicaciones como el almacenamiento de datos, en parte porque ninguna de ellas puede proporcionar una capacidad de bases de datos distribuida.

Los productos propietarios *Oracle*, *DB2*, *Ingres*, *Informix*, *Progress*, *Mimer* y *Sybase* valen para funcionar con GNU/Linux y podrían considerarse opciones preferentes para aplicaciones pesadas de bases de datos donde los productos OSS aún no resultan adecuados. Las herramientas de desarrollo de *Oracle* están soportadas GNU/Linux.

## 12. Aplicaciones – Grupos secundarios

### 12.1. Sistema Operativo

Hay varios sistemas operativos OSS, como *OpenBSD*, *FreeBSD* y *NetBSD*, y varias “Distribuciones” (se explica más adelante) de GNU/Linux, aunque mucha gente sólo haya oído hablar de GNU/Linux y por lo general sólo por el nombre de Linux.

Un sistema operativo consiste en un kernel que funciona en modo supervisor, junto con programas de apoyo que funcionan a su vez bajo el control de un kernel en modo usuario. Linux es un kernel pero necesita cargadores, compiladores, programas controladores (*drivers*), etc. Muchos de estos programas secundarios proceden del proyecto GNU de la Free Software Foundation y por eso todos deberían llamarse GNU/Linux que es el término utilizado en este informe.

El kernel *Linux* se da en un paquete con un conjunto de programas y aplicaciones de apoyo procedentes de una serie de empresas como *Red Hat*, *SuSE* y *Mandrake* como Distribución. Los contenidos de una Distribución deberían poder interactuar, y el kernel puede muy bien ser “parcheado” con cambios no disponibles en otras Distribuciones. Por ello se puede considerar la elección de Distribución, ya que cada una tiene sus puntos fuertes y débiles.

Hay otras Distribuciones como *Debian* y *Gentoo* que no están preparadas por una empresa comercial y esto tiene ciertas implicaciones por el modo en que se da apoyo. El apoyo para estas distribuciones procede bien de terceros bien del acceso a listas de correo en Internet. Tanto en un caso como en otro pueden prestar niveles de cobertura aceptables.

*Debian* es apreciada por su solidez y su sección estable contiene un código que ha sido testado a fondo por mucha gente en todo el mundo. También hay otras dos secciones que dan crecientes niveles de software casi de vanguardia. La rama estable también tiene fama de no estar actualizada. Esto es injusto en cierta medida porque a la mayoría de los usuarios comerciales lo que más les interesa es la estabilidad y la ausencia de errores, y no se puede dar apoyo al último periférico que haya salido al mercado.

*Gentoo* es una distribución sólo de fuentes, lo que significa que la administración puede construir fácilmente sus propios binarios, adaptando la Distribución a su entorno y hardware. Construir una distribución de este tipo a partir de cero lleva tiempo pero una vez que los binarios se han construido están generalmente disponibles. Esta es una nueva distribución y vale la pena estudiarla. Dado que la mayoría de las demás Distribuciones se suministran con un código total de fuentes, es posible adaptar cualquiera de ellas de la misma manera. Sin embargo, *Gentoo* puede ser más sensible a ese tratamiento.

Las Distribuciones comerciales vienen en diferentes paquetes con diferentes niveles de apoyo disponibles. La Distribución disponible por Internet cuenta invariablemente con apoyo sólo un año; así pues se espera que los usuarios instalen una versión mejorada. La mayoría de las empresas proporcionan una versión “de empresa” con un apoyo garantizado de cinco años o más, que parte de versiones estables. Esas versiones también tienen un contrato de apoyo que a veces recibe el nombre de licencia, incluso aunque el código se dé con licencia mediante GPL y LGPL, y no pueda hacerse de otro modo. Y es el disponer de esas distribuciones estables y con apoyo lo que muchas Administraciones quieren. De hecho, una de las razones para pasarse al OSS es la ausencia de presión para mejorar instalar versiones nuevas constantemente y sin necesidad alguna. Las empresas prometen introducir las correcciones de los errores. Red Hat, por ejemplo, tiene su Gama

de Empresa que consiste en tres productos, dos para servidores y uno para puestos de trabajo técnicos, y cada uno de ellos parece basarse en la versión 7.3 de *Red Hat Linux*, aun cuando la versión actual que se puede descargar es la 9.

Nosotros opinamos que GNU/Linux es la plataforma preferida para los puestos de trabajo pues ofrece mejores herramientas de configuración y una variedad de configuraciones de empaquetamiento más adecuadas para usar en el escritorio. Además, algunos productos para escritorio muy conocidos no trabajan con todas las alternativas (El navegador de web *Mozilla* actualmente no trabaja en *OpenBSD*).

En cuanto a los servidores, la situación está mucho menos clara. *OpenBSD* tiene con mucho el mejor historial de seguridad de un sistema operativo OSS, y sólo se le puede achacar una vulnerabilidad explotable a distancia en 6 años. Debería ser la plataforma preferida para todo lo que requiera una seguridad por encima de la media (como los cortafuegos y los servidores de las zonas desmilitarizadas).

Las aplicaciones para servidores suelen funcionar bien en todas las plataformas BSD y en GNU/Linux, aunque muchas se han creado para GNU/Linux y están por lo tanto portadas. El software propietario a menudo suele estar disponible sólo sobre GNU/Linux.

## **12.2. Interfaz de usuario**

### **12.2.1. Gestor de escritorio – aspecto visual y operacional**

Hay varias opciones, que van de los gestores de ventanas ligeros, sencillos, como *icewm*, a los gestores de sesiones totalmente desarrollados, como los incluidos en *Gnome* y *KDE*. La opción dependería del uso al que se va a destinar.

De los gestores de sesiones el de *KDE* es el más maduro, pero *Gnome* está a punto de darle alcance. *Gnome* tiene el apoyo de Sun Microsystems y los miembros de la Fundación Gnome. **netproject** considera que tiene mejor arquitectura y cree que su futuro es más prometedor.

Para cualquier Administración la elección se haría según las preferencias personales; los dos entornos son muy capaces. Las aplicaciones creadas para funcionar en un entorno también lo harán en el otro si bien algunas características concretas, como la gestión de sesiones, podrían no trabajar bien.

### **12.2.2. Idioma**

Los gestores de escritorio ofrecen la mayoría de los idiomas europeos pero deben “parchearse” en su apoyo a algún idioma concreto.

## **12.3. Seguridad**

Todos los grupos funcionales deben ser configurados teniendo siempre presente la cuestión de la seguridad. La seguridad en el software sólo funciona si existe en el marco más amplio de la gestión de la seguridad. **netproject** no ha investigado a fondo todas las funciones dadas como parte de este informe.

### **12.3.1. Cifrado de los datos**

#### **12.3.1.1. Datos en tránsito**

Los datos confidenciales sobre las LAN deberían estar cifrados siempre que sea posible. Los datos sensibles sobre Internet también deberían estar cifrados. Esto se puede hacer con un “túnel” en las conexiones sobre productos como *ssh* y *stunnel*.

#### **12.3.1.2. Datos en memoria**

Los datos confidenciales que se guardan en dispositivos móviles deberían cifrarse en disco. Lo ideal sería que todos los datos se cifren pero esto impondría significativas sobrecargas que no siempre resultarían aceptables. Hay un cierto número de sistemas de archivo seguros y el próximo kernel de Linux proporciona mayor apoyo para ellos. Por ejemplo, en

<http://koeln.ccc.de/archiv/drt/crypto/linux-disk.html> se habla de los distintos métodos disponibles.

#### **12.3.2. Autenticación**

Los métodos seguros para identificar únicamente a una persona o una máquina son parte de la comunicación con otras personas o máquinas. Esto incluye las firmas y las infraestructuras PKI. No se testó ningún sistema PKI como parte de este proyecto. Toda la autenticación se hizo sobre las bases de datos LDAP con una contraseña estándar.

#### **12.3.3. Autorización**

Tras la autenticación, hay que determinar lo que una persona o una máquina pueden hacer y en qué circunstancias. Esto suele ser parte de sistema operativo o el código de aplicación. El Control de Acceso Basado en el Papel (*RBAC*, o *Role Based Access Control*) ha sido definido por el NIST de Estados Unidos y está disponible para Linux. (Véase <http://csrc.nist.gov/rbac/>).

#### **12.3.4. Control de virus**

El control de los virus es necesario sobre todo para detener la transmisión de virus a otros sitios no OSS. Aunque el correo electrónico es uno de los principales medios de transmisión de virus, no es el único y por eso hay que generalizar el escaneo de archivos para detener la transmisión por otros medios.

Lamentablemente, no conocemos ningún producto OSS que haga ese escaneo. Pero configurando bien los sistemas de archivo en el servidor y en los escritorios se puede garantizar que los únicos archivos ejecutables son los que fueron instalados por el administrador del sistema. Por eso es importante que los administradores del sistema se aseguren de que los archivos que instalan sean fiables, por ejemplo comprobando la firma del vendedor de la Distribución que hay en los archivos.

#### **12.3.5. Servidor proxy**

Hay disponibles distintos servidores proxy de OSS inteligentes o semi-inteligentes.

De los servidores proxy en la web, *squid* es el más popular. Tiene un producto hermano (*squidguard*) que evita el acceso a una lista de sitios web prohibidos.

#### **12.3.6. Cortafuegos**

Todos los sistemas operativos OSS actuales tienen cortafuegos internos de filtrado de paquetes, la mayoría de los cuales son de estados. Los cortafuegos de estados son los que mantienen información sobre las conexiones en curso y las corrientes de datos a través del cortafuegos, y permiten pasar a los paquetes que están relacionados con dichas conexiones, mientras que filtran o rechazan a los paquetes que no lo están. Los cortafuegos que no son de estados examinan cada paquete por sus propios méritos, sin guardar ningún registro de los paquetes anteriores. Existen accesorios especializados para protocolos como los ftp y de telefonía H.323 que usan medios de conexión no estándar.

*iptables*, actualmente incluido en el GNU/Linux, e *ipfilter*, incluido con *FreeBSD*, son buenos productos cortafuegos. *Packetfilter* se incluye ahora con *OpenBSD* y también está bien considerado. La buena práctica en el caso de los cortafuegos externos es tener dos diferentes entre la conexión a la red pública y los servidores externos. Pero nosotros no recomendamos un ejemplo único.

### 12.3.7. Redes privadas virtuales (VPN)

#### 12.3.7.1. OpenVPN

Disponible para la mayoría de los elementos de Unix, es una oferta potente y madura. Entre sus características están el cifrado de clave pública, la compresión dinámica para la gestión del ancho de banda y la capacidad para trabajar con NAT (Network Address Translation, o Traducción de direcciones en la red). Para más información, consulte también <http://openvpn.sourceforge.net/>

#### 12.3.7.2. FreeSWAN

Es una aplicación GNU/Linux de los estándares IPSEC e IKE, lo que significa que operarán recíprocamente con los dispositivos compatibles como, por ejemplo, encaminadores especiales y otros sistemas operativos. Dado que el IPv6 da apoyo a IPSEC originariamente, *FreeSWAN* podría preferirse si se usa este nuevo estándar. Para beneficiarse de la extensión exclusiva de “Cifrado oportunista”, que puede automatizar la seguridad, los registros del DNS deben actualizarse, lo que podría implicar ciertas limitaciones. **netproject** entiende que IPSEC también puede tener problemas con NAT. Para más información, véase también <http://www.freeswan.org/>.

#### 12.3.7.3. CIPE

Este es menos maduro que los otros dos y el apoyo de clave pública está aún en fase de experimentación. Aunque sí funciona con NAT, está disponible para *Windows*, y va bien con *Red Hat Linux* (uno lo puede configurar con su intuitiva herramienta de Control de Dispositivos de Red). Más información en <http://sites.inka.de/~W1011/dev/cipe.html>.

## 12.4. Gestión

El sitio <http://www.infrastructures.org/> proporciona información detallada sobre cómo gestionar una red de máquinas, tanto servidores como escritorios, y tiene ciertas herramientas de OSS para una serie de tareas de mantenimiento del sistema. Lo mantiene alguien que ha realizado esas tareas durante muchos años. **netproject** está de acuerdo con prácticamente todo el contenido excepto, en principio, con la parte dedicada a la gestión de las impresoras.

En el citado sitio se muestra que Unix, y por extensión GNU/Linux, tiende a realizar la gestión mediante herramientas que se juntan a partir de unidades más sencillas de una sola función. Este planteamiento modular es extremadamente poderoso y es lo que permite que los administradores de los sistemas Unix y GNU/Linux sean más eficaces y efectivos. También significa que el mercado de los kits de herramientas es pequeño, ya que cada uno de los administradores de sistemas tiende a construir su propio juego de herramientas.

Hay herramientas propietarias a disposición del usuario, como *Tivoli* de IBM y *Unicenter* de Computer Associates.



### 12.4.1. Gestión de usuarios

El mantenimiento de usuarios y grupos de usuarios, incluida la gestión de las contraseñas. Productos como *Directory Administrators* y *gq* permiten el mantenimiento de las bases de datos de LDAP.

### 12.4.2. Gestión de configuraciones

Aunque un cliente gestionado centralmente y bien diseñado sólo debería tener un estado local mínimo, actualizar su configuración sin volver a instalarla desde cero sigue siendo algo deseable para las grandes redes de las que se espera que estén activas mucho tiempo. Por ejemplo, si un servicio central fundamental se cambia, es posible que los clientes tengan que ser reconfigurados para poder usarlos.

#### 12.4.2.1. Mantenimiento manual de la configuración

Los Administradores pueden mantener las actualizaciones de una configuración de forma manual ya que pueden dotar de software a las actualizaciones. Sin embargo, aparecen problemas de sincronización similares. La modificación manual de los archivos de configuración, con frecuencia guardados en archivos de texto plano, son especialmente propensos a los errores mecanográficos

#### 12.4.2.2. Cfengine

El *GNU Configuration Engine* (se puede consultar <http://www.dfengine.org/>) automatiza la configuración remota de los clientes integrados en la red. Da apoyo a una gran variedad de elementos de UNIX, y su potente concepto de clase permite gestionar a diferentes grupos de clientes con una mínima configuración. Agentes autónomos para los clientes pueden mantener los archivos de texto, las interfaces de red, los enlaces y los permisos relacionados con los archivos, el almacenaje temporal y los sistemas de archivos montados.

Las siguientes son algunas de las funciones que pueden automatizarse mediante *cfengine*:

- Controlar y configurar la interfaz de red.
- Editar archivos de texto.
- Hacer y mantener enlaces simbólicos, incluidos los enlaces múltiples a partir de una orden sencilla.
- Comprobar y establecer los permisos y la propiedad de los archivos.
- Limpiar (borrar) los archivos basura que desordenan el sistema.
- Montaje automatizado, sistemático, de sistemas de archivos (en Unix).
- Comprobar la presencia de importantes archivos y sistemas de archivos.
- Ejecución controlada de comandos (órdenes) de shell y *scripts* de usuarios.

*Cfengine* sigue una estructura de decisiones basada en las clases.

#### 12.4.2.3. Configurador del sistema

*System Configurator* (<http://sisuite.org/systemconfig/>) forma parte de la *System Installation Suite*, y lo usa el *System Installer*. Puede configurar y mantener muchos componentes de una instalación de GNU/Linux a través de muchas distribuciones como puesta en red, almacenaje, zona temporizada y arranque.

### 12.4.3. Gestión del software

Esta sección se ocupa del sistema de mantenimiento de los clientes desde la configuración inicial sobre nuevo hardware a las actualizaciones permanentes del software y la configuración, y de algunas tecnologías disponibles para facilitar su gestión.

#### 12.4.3.1. Instalación del software

La instalación del sistema es la implantación inicial del software y la configuración necesarios para una máquina. Es posible que las máquinas salidas de fábrica no tengan ningún sistema operativo, o lleguen con el software preinstalado. Las máquinas más antiguas que tienen software no deseado pueden reutilizarse si se les vuelve a instalar un sistema reciente en lugar del que tenían.

La primera tarea de un instalador de sistemas es arrancar el sistema operativo en la máquina objetivo. En el caso de que esto no se pueda hacer, como las máquinas que salen de fábrica con discos duros no inicializados, el BIOS debe tener por lo menos un método de reinicio aparte del disco duro. El método antiguo consiste en arrancar desde un disquete, y aunque esto es muy corriente presupone la existencia de una unidad de disquete. Y es un método desfasado. Los disquetes son lentos, poco fiables y ofrecen muy poco espacio para el software de instalación del sistema de acuerdo con los estándares modernos. La mayoría de las máquinas construidas a partir de 1997 se pueden arrancar desde un CD-ROM emulando el sector de arranque del disquete. Si hay unidad para CD, es más rápida y ofrece más espacio tanto para el software de arranque inicial como para cualquier software posterior necesario. El método de arranque más sofisticado es el arranque desde la red. No todos los soportes lógicos incorporados o tarjetas de red para BIOS valen para esta característica nueva. El Entorno de Pre-Ejecución (PXE) forma parte del estándar comercial *Wired for Management* (WfM), y permite que la mayoría de las máquinas compradas a partir de 1998 arranquen desde la red local.

El instalador debe acceder a los medios de instalación adecuados que contengan el software de mayor nivel que se pueda ejecutar una vez que la máquina haya arrancado. Normalmente este software estará guardado en un CD-ROM local o en un servidor de archivos en red. Se puede usar un simple disco compacto para guardar una imagen de software, y la capacidad de un CD-ROM debería bastar para un escritorio básico de Administración que utilice una compresión de archivos. Esta imagen estática podría valer si el software no se va a cambiar, o si sólo se necesita una instalación básica estable para añadir más software. En general, una instalación por red es más flexible, puede ser más rápida, ofrece mayor capacidad y se adapta mejor para instalaciones paralelas y múltiples que si hay que compartir discos de instalación entre los clientes

El instalador del sistema transfiere el software desde el medio elegido a una unidad de disco duro local situada en la máquina objetivo, y la prepara para arrancar. Aquí entra la detección del hardware, la comprobación de la capacidad del disco y la configuración de los pormenores de la red.

A continuación se habla de los posibles métodos de instalación:

#### 1. Instalación manual

La instalación más básica es la que hace un administrador de sistemas. Lo normal es que el software se facilite en discos compactos, incluido un disco de

instalación "arrancable". Algunas instrucciones automáticas pueden guiar al administrador, pero al final toda personalización es manual. Como todos los detalles de la red, la configuración del hardware, la partición del disco duro y la selección del paquete se deben introducir manualmente, este proceso lleva tiempo y es propenso al error humano. La mayoría de las Distribuciones tienen su propio programa de instalación, como por ejemplo el *anaconda* de Red Hat y el *YAST2* de SuSE.

## 2. Clonado de imágenes

Si los clones casi idénticos resultan adecuados, se puede instalar manualmente un cliente patrón y luego reproducirlo. Las distribuciones en conexión como *Knoppix* (que arrancan un entorno completo de GNU/Linux a partir de un único CD-ROM - véase <http://www.knopper.net/knoppix/>) y otros discos de rescate se pueden usar para copiar las imágenes de los sistemas de archivos del cliente patrón en otras máquinas. La configuración y la personalización se pueden añadir mediante *scripts* antes o después de la instalación. Dado que los sistemas de archivos en bruto, y no sólo los archivos que contienen, se pueden copiar en disco, esto da el mejor tiempo de instalación posible. No obstante, configurar clones no idénticos es menos eficaz y exige la habilidad de un experto.

## 3. Instalación totalmente automática

*FAI* (<http://www.informatik.uni-koeln.de/fai/>) instala de forma automática la distribución *Debian*. A los paquetes de software se accede desde un sitio *Debian*, que se puede replicar localmente para obtener velocidad o personalización. El kernel de instalación proporcionado se puede arrancar desde la red o el disquete, pero el arranque desde un CD-ROM aún está en preparación. Aunque *FAI* se diseñó para una reproducción idéntica de máquinas agrupadas, el software *cfengine* que se describió anteriormente se utiliza para la configuración del sistema y permite una amplia flexibilidad si es necesaria.

## 4. System Imager (Generador de imágenes del sistema)

*System Imager* (<http://systemimager.org/>) proporciona instalación del sistema, configuración y mantenimiento a grandes redes de máquinas, de preferencia con un hardware similar, a través de varias distribuciones. Puede arrancar desde un disquete, un CD-ROM o servidores de red PXE. Tanto las instalaciones *Debian* como *Red Hat* se han testado, pero el software de *System Configurator* usado tiene por objeto dar apoyo a las distribuciones GNU/Linux.

Un cliente patrón se instala y configura manualmente. Sus sistemas de archivos se replican a un servidor de imágenes, desde el que se instalan en las máquinas objetivo. Si el cliente patrón está actualizado, estos cambios se propagan a clientes replicados usando *rsync*. Aunque *rsync* envía mínimamente diferencias de archivos a red, puede necesitar bastante memoria para hacerlo. Como las modificaciones se refieren al cliente patrón, *System Imager* es lo más adecuado para clientes objetivo con un hardware idéntico o muy similar.

## 5. Kickstart de Red Hat

*Kickstart* (<http://www.tldp.org/HOWTO/kickStart-HOWTO.html>) es el software de instalación automática de Red Hat. Instala las distribuciones

Red Hat desde un CR-ROM, un disco duro o la red, y arranca igualmente desde la red, un CD o un disquete. El instalador *anaconda* ofrece interfaces gráficas o de texto, y puede ser interactivo o automatizado totalmente por un archivo de configuración. El software de detección de hardware *kudzu* atiende automáticamente a una amplia gama de dispositivos. Las opciones generales de instalación se pueden incluir en el archivo de configuración, y las ampliaciones se pueden añadir con *scripts* de pre y post instalación.

Con su inteligente software de detección y configuración, *kickstart* se puede usar para automatizar instalaciones similares a través de toda una variedad de objetos de software. La selección de paquetes desde una distribución Red Hat estándar es sencilla, pero las actualizaciones o ampliaciones también se pueden incluir personalizando el proceso *kickstart*.

#### **12.4.3.2. Mantenimiento del software**

Las instalaciones de software no permanecen estáticas a lo largo de toda su vida. Tras la instalación inicial se harán actualizaciones de software del tipo de corrección de errores o de seguridad. Además, será necesario quitar o añadir paquetes para gestionar el software sin reinstalar todo un sistema.

Siempre que sea posible, las actualizaciones deberían hacerse utilizando las técnicas de recibir (*push*) en vez de obtener (*pull*). La decisión de obtener actualizaciones debería tomarla la máquina, sea servidor o escritorio, una vez que se haya comprobado a sí misma a través de un servidor maestro. Las actualizaciones no deberían estar bajo el control de los usuarios. De esta manera, las máquinas se pueden mantener en el mismo nivel de revisión.

#### **1. Mantenimiento manual del software**

Los administradores de sistemas pueden hacer las actualizaciones de software de manera manual. Ello podría implicar el conectarse al cliente a distancia, copiando los paquetes actualizados e instalándolos en el gestor de paquetes original del distribuidor. Pero aunque esto ofrece un correcto control del administrador, es proclive a los errores y hace difícil la sincronización de grandes grupos de máquinas. Algunas distribuciones ofrecen herramientas de actualización para mantener sus paquetes estándar, pero siguen necesitando una intervención manual y es posible que no proporcionen ampliaciones a la distribución básica.

#### **2. Red Carpet de Ximian**

*Red Carpet* (<http://www.ximian.com/products/redcarpet/>) es un software libre de actualización de Ximian. Comenzó como gestor de paquetes gráficos para el software de escritorio de Ximian, pero ahora ofrece acceso seguro a líneas de comandos remotos y más canales de software como, entre otros, actualizaciones de distribuciones. *Mandrake*, *SuSE* y *Red Hat* tienen su apoyo actualmente. Ofrece automatización y administración remotas, por lo que un gran número de clientes se pueden mantener centralmente. Pero conserva alguna rémora de su diseño original. No da apoyo al kernel ni a las actualizaciones de arquitectura mejorada. Puede configurarse para actualizar software procedente de canales personalizados. Se puede usar un producto de servidor propietario, *Red Carpet Enterprise*, para facilitar la gestión de grandes grupos de software.

#### **3. Enterprise Network (Red para empresas) de Red Hat**

Red Hat ofrece una serie de servicios para actualizar software como parte de su *Enterprise Network* propietaria ([http://www.redhat.com/software/rhen/software\\_delivery](http://www.redhat.com/software/rhen/software_delivery)) El más potente es su *Satellite Server*, que permite la plena personalización de actualizaciones y erratas. Todos los servidores funcionan con sus clientes estándar *Update Agent* en la distribución. Y lo mismo dicho en cuanto a permitir el uso de la interfaz gráfica se puede aplicar aquí al *Red Carpet* antes comentado.

#### 4. APT de Debian

*APT* es un conjunto (suite) estándar de herramientas que se suministra con la distribución *Debian GNU/Linux* que permite actualizaciones automáticas del software instalado en una máquina. Puede comprobar las dependencias entre los paquetes de software instalados en la máquina y disponibles desde los repositorios de software a los que su configuración le permite controlar, y recuperar e instalar importantes actualizaciones disponibles en un repositorio. Las empresas pueden configurar e instalar sus propios repositorios de software para instalarlos en sus clientes (*Debian* incluye herramientas para configurar y mantener esos repositorios), usar repositorios proporcionados por *Debian* y por otros, o usar cualquier combinación de estas fuentes de software actualizado. *APT* se ha hecho funcionar en sistemas operativos como *Red Hat Linux* y *Mandrake*, donde ofrece una funcionalidad similar a *Red Carpet*, y en algunos casos incluso mejor.

#### 12.4.4. Gestión del hardware y supervisión del sistema

El hardware se puede supervisar para ver si tiene defectos y posibles fallos, por ejemplo usando discos duros habilitados SMART y hardware que comprueba la salud del sistema. Los sistemas de hardware y software también se supervisan para ver si tienen fallos, posibles fallos, ausencia de servicio y falta de capacidad.

##### 12.4.4.1. MRTG y Snmpd

*MRTG* (*Multi-Router Traffic Grapher* - Esquemizador de tráfico con enrutador múltiple, <http://people.ee.ethz.ch/~oetiker/webtools/mrtg/>) es una herramienta de supervisión diseñada en principio para hacer el seguimiento y trazar el gráfico del uso de la capacidad en los enlaces de la red. No obstante, ha pasado a ser una herramienta capaz de hacer un seguimiento virtual de cualquier parámetro cambiante, y puede usarse para supervisar variables tales como el uso del espacio de disco y memoria, el uso de servicios en red (por ejemplo, los referidos a la estadística sobre volúmenes de correo electrónico procesados, páginas web servidas, etc.), la temperatura del sistema y la velocidad del ventilador, y otras variables.

*Snmpd* (*Simple Network Management Protocol Daemon*, <http://net-snmp.sourceforge.net/>) es un servidor de gestión de sistemas que puede ejecutarse en cada máquina de escritorio de una empresa. Proporciona información sobre la gestión de los sistemas a los clientes, normalmente a un cliente central de SNMP que reúne las estadísticas de diversas máquinas. *MRTG* puede actuar como ese cliente y realizar esta función, proporcionando una visión general gráfica del estado de un gran número de máquinas cliente.

#### 12.4.4.2. Nagios

*Nagios* (antes conocido como *NetSaint*, <http://www.nagios.org/>) es un anfitrión que se puede personalizar, un sistema de gestión y supervisión de redes y servicios. Puede supervisar servicios de red y realizar diversos procedimientos de recuperación si descubre que un servicio no está disponible o tiene problemas, incluido el de llamar a *scripts* de recuperación automática y administradores de sistemas de alerta para el programa. *Nagios* también puede proporcionar informes y visiones generales del estado presente y pasado de los servicios que supervisa.

#### 12.4.4.3. smartd

El conjunto de herramientas *SmartMonTools* (<http://smartmontools.sourceforge.net/>) incluye un Daemon (duende, demonio) llamado *smartd* pensado para supervisar la función SMART (*Self-Monitoring, Analysis and Reporting Technology* - Tecnología para la creación de Informes, el Análisis y la Auto-supervisión), o modernas unidades de disco duro. Como estos dispositivos son uno de los componentes que más fallan en un ordenador moderno, SMART está destinado a supervisar los parámetros de la unidad y advertir a un administrador del sistema de los posibles fallos antes de que se produzcan. *smartd* está pensada para recibir esas advertencias y tomar medidas, normalmente alertando al administrador del sistema.

### 12.4.5. Gestión de la impresora

#### 12.4.5.1. LPRng

*LPRng* (<http://www.lprng.com/>) es una implementación desarrollada activamente del antiguo sistema *lpr/lpd* de estándar BSD. Contiene un cierto número de mejoras que pueden hacerlo mucho más robusto y fácil de gestionar que los productos originales. Al autor le interesa sobre todo que *LPRng* sea seguro. Hasta hace poco esta era probablemente la elección para la gestión de la impresora pero en los últimos tiempos *CUPS* ha avanzado mucho y su elección es ahora menos cara.

#### 12.4.5.2. Sistema Común de Impresión Unix

El Sistema Común de Impresión Unix o *CUPS* (<http://www.cups.org/>) está pensado como sistema de impresión Unix listo para la empresa. Se basa en el Protocolo de Impresión Internet estándar o IPP, e incorpora una función de navegador que permite la distribución automática por la red de los detalles de los nombres y características de las impresoras. *CUPS* incorpora también una interfaz de usuario basada en la web para administrar y configurar las impresoras. Hay programas controladores disponibles para la mayoría de las impresoras.

#### 12.4.5.3. Kprint y GnomePrint

*KDE* y *Gnome* incorporan sus propios subsistemas de impresión aptos para definir las aplicaciones de usuario en la mayoría de los sistemas de cola de impresión de uso más corriente, *LPRng* y *CUPS* incluidos.

### 12.5. Copia de seguridad y recuperación

Se considera que todos los datos de la Administración y los usuarios están en uno o más servidores. Es necesario poder hacer volcados incrementales, encontrar volcados con archivos dados y restaurar los archivos individuales o los sistemas de archivos en su



totalidad. La copia de seguridad de los datos del usuario tiende a ser más fácil en los sistemas Unix y OSS que con *Windows* porque los archivos de datos de usuarios, incluidos los datos de configuración, suelen estar en un único directorio. Esta es otra área en la que un producto propietario como *Legato* puede ser necesario para obtener las características y el fino control requerido en un sitio grande.

### 12.5.1. Dump y Restore

Estos dos programas se entregan como parte de la mayoría de las distribuciones y veces se usan junto con *tar* y *cpio* en *scripts* personalizados para hacer copias de seguridad y recuperar máquinas únicas.

### 12.5.2. Amanda

*Amanda* (<http://www.amanda.org/>) es un producto de servidor de cliente diseñado para respaldar múltiples máquinas en un solo dispositivo. También puede respaldar a *Windows* a través de *Samba*.

## 12.6. Otros servicios

### 12.6.1. Servidores temporales

En un entorno altamente en red es fundamental que todas las máquinas – tanto servidores como escritorios – tengan la misma noción del tiempo actual. Uno o más servidores se designan como servidores maestros y obtienen su tiempo bien de un reloj adjunto bien de servidores de tiempo externos en Internet. Todas las demás máquinas son esclavos que se sincronizan según esos maestros.

La sincronización del tiempo se puede hacer ejecutando *ntp* (<http://www.ntp.org/>) en las máquinas, y ello permitir mantener una red de máquinas en el rango de un segundo con bastante facilidad.

*Chrony* (<http://goto/chrony/>) es una alternativa a *ntp*. Tiene algunas características que lo hacen más adecuado para nodos NTP de mayor estrato que *ntp*, mientras *ntp* es mejor para los nodos de bajo estrato que pueden tener que intercambiar datos directamente con cosas como receptores de GPS y relojes atómicos. También hay productos OSS para *Windows* que resultan útiles en un entorno mixto, como *Automachron* y *nettime* - en <http://go.to/chrony/> se da más información sobre los dos.

### 12.6.2. Servidores de infraestructuras de red

Estos son los servicios necesarios para ejecutar una red basada en los TCP/IP.

#### 12.6.2.1. Encaminamiento

Los encaminadores permiten que una red grande se divida en varias más pequeñas conectadas entre sí. Los encaminadores tienen la tarea de dirigir los paquetes de una sub-red a otra para que lleguen a su destino. Construir encaminadores requiere un buen entendimiento de los protocolos básicos y es posible que muchas Administraciones quieran comprar encaminadores propietarios.

Sin embargo, para quienes deseen construir sus propios productos existen *Bird* (<http://bird.network.cz/>) y *GNU Zebra* (<http://.zebra.org/>).

#### 12.6.2.2. DNS

Una red de TCP/IP necesita algunos medios para traducir las direcciones de IP a nombres de dominio con significado humano y viceversa. El DNS es un protocolo

con un cierto número de servidores en intercomunicación cada uno de los cuales contiene datos. El DNS es fundamental para trabajar en Internet. Hay varios programas para construir los servidores DNS como, entre otros, *BIND* (<http://isc.r/products/BIND/>), *MaradNS* (<http://maradns.org/>) y *MyDNS* (<http://mydns.bboy.net/>). *Bind* es el que más se usa.

### 12.6.2.3. DHCP

DHCP es un protocolo escrito en <http://www.dhcp.org/> que permite que las máquinas obtengan sus datos de red en tiempo de arranque a partir de un servidor central o varios. El DHCP permite el uso eficaz de direcciones de IP escasas o raras y reasignará las direcciones siempre que sea posible, También permite la administración central de muchas direcciones globales como servidores de portales y nombres. El producto principal procede de <http://www.isc.org/products/DHCP/> y consiste en una aplicación de cliente y servidor. El cliente tiene que trabajar en todas las máquinas cliente participantes. Los dos productos vienen con la mayoría de las distribuciones estándar.

### 12.6.3. Servidores de archivos

Los servidores de archivos en red permiten a las máquinas que funcionan en la red su acceso al almacenamiento de datos en una máquina remota como si fuera local.

#### 12.6.3.1. NFS

Este es el estándar de facto y ha estado en uso durante muchos años. El subset que se implanta normalmente no proporciona una gran seguridad, aunque hay una variante segura que se ha puesto en marcha en algunas variantes comerciales de Unix.

NFS consiste en un servidor que exporta archivos desde la máquina en la que está funcionando a clientes que funcionan en otras máquinas adjuntas a la red. Se controla lo que otras máquinas puedan adjuntar a esos archivos pero una vez que se adjunta uno entonces el tráfico en la red queda fuera de toda sospecha. En la versión de Linux hay una mínima autenticación de los usuarios de Linux.

#### 12.6.3.2. Samba

*Samba* es un producto que pone en marcha el protocolo SMB de Microsoft. Para una descripción más detallada, véase el punto 14.5.1. Es fundamental para la integración de los sistemas basados en Windows y el OSS, y viene en la mayoría de las distribuciones estándar. Su uso se describe con más detalle en el Capítulo 14.

#### 12.6.3.3. Netatalk

En el caso de los que tienen máquinas Apple de Macintosh, *netatalk* proporciona la ejecución del protocolo AppleTalk. (Véase <http://netatalk.sourceforge.net/>).

#### 12.6.3.4. OpenAFS, CODA e Intermezzo

Estos productos ejecutan un sistema de archivos distribuidos a diferentes grados. Con ese sistema el acceso a los archivos puede continuar mientras la red falla, pues la memoria cache hace que parezca que está conectada. No se trata de un problema trivial y los productos lo resuelven de diferentes formas. Este tipo de sistema de archivos es realmente necesario en el caso de los laptops o las máquinas unidas a una conexión transitoria. El otro modo de proporcionar la



misma funcionalidad es tener un almacenamiento local sincronizado periódicamente con un servidor central. Para más información sobre cada producto, véase <http://www.openafs.org/>, <http://www.coda.cs.cmu.edu/> y <http://www.inter-mezzo.org/>.

Por su parte, en <http://www.inter-mezzo.org/docs/bottlenecks.pdf> se tratan a fondo las características de todos ellos.

#### **12.6.4. Servicios de directorio**

Proporcionan una forma rápida de consulta de nombres, direcciones y datos afines.

El estándar más popular para servicios de directorios es el LDAP. Consiste en un protocolo abierto y se incluye en muchos productos como, por ejemplo, *Evolution* y *OpenOffice.org*. El LDAP trabaja con definiciones de datos llamadas esquemas y las administraciones pueden crear sus propios esquemas personalizados. Lamentablemente, los esquemas utilizados por las aplicaciones no siempre son compatibles entre sí, lo que significa, por ejemplo, que a *OpenOffice.org* le resulta difícil leer los datos de *Evolution* y viceversa.

La aplicación OSS *OpenLDAP* se ajusta al estándar LDAPv3, y la versión 2.1 y posteriores se puede configurar con toda una serie de gestores de bases de datos (como archivos planos, SQL o según el sitio).

La mayoría de los groupware proporcionan alguna forma de servicio de directorio, pero pocos son los que se integran de manera compatible en el LDAP. Con pocas facilidades para las técnicas de "cortar y pegar", es difícil usar la base de datos de contactos que proporcionan en agentes de correo externo. La mayoría de ellos ofrecen sus propios agentes de correo, pero no dicen mucho en cuanto al nivel de integración disponible en el gestor de contactos incorporado.

*OpenOffice.org*, *Evolution* y *Mozilla* proporcionan funciones integrales de libreta de direcciones. Sin embargo, los formatos de almacenamiento utilizados no son intercambiables. Para poder interactuar es necesario tener algún sitio a medida.

#### **12.6.5. Soporte de sistemas heredados**

##### **12.6.5.1. Emulación por terminal**

El uso de *xterm* con una configuración de variables de entorno TERM puede emular la mayoría de los tipos de terminales basados en caracteres, como por ejemplo VT220 y VT100. Hay un emulador 3270 específico llamado *x3270*. En los productos propietarios se pueden encontrar emulaciones basadas en página.

##### **12.6.5.2. Visualización remota**

El la sección 13.3 se trata este tema.

##### **12.6.5.3. Emulación**

El la sección 13.4 se trata este tema.

## 13. Migración de aplicaciones – Visión general

Una vez que se ha elaborado una lista de las aplicaciones, éstas se pueden agrupar en las siguientes categorías:

### 13.1. Aplicaciones propietarias que tienen un equivalente OSS

Algunas aplicaciones, como *Office*, *Lotus Smart Suite*, *WordPerfect*, *Framemaker*, *Quark Express* y *Photoshop*, tienen equivalentes que funcionan originariamente con OSS, incluidas *OpenOffice.org*, *Gnumeric*, *Evolution* y *The GIMP*. En este caso, el producto OSS ha de ser testado para asegurarse de que procura la necesaria funcionalidad.

### 13.2. Aplicaciones propietarias que funcionan en un entorno OSS

Algunas aplicaciones, como *Acrobat Reader* de Adobe, tienen una versión que funciona originariamente con OSS. Si no hay alternativa OSS a la aplicación, entonces es necesario asegurarse de que las características requeridas se ejecutan en la versión propietaria. Si existe una alternativa OSS y se puede hacer una migración parcial, entonces se ha de hacer la elección a partir de las características que ofrezcan las versiones de OSS y las propietarias.

### 13.3. Software al que se puede acceder por visualización remota

Otro planteamiento es el que contempla ejecutar las aplicaciones en un servidor y trasladar la visualización al escritorio; este es el planteamiento del cliente ligero. Productos como *Windows Terminal Server*, *Citrix* y *Graphon* permiten que las aplicaciones se ejecuten en un servidor que tenga *Windows* de manera multiusuario. Esto significa que una aplicación escrita para funcionar en el escritorio en modo de usuario único podría tener que ser alterada para funcionar con estos productos. Esto no será posible sin un código fuente y los vendedores terceros quizá no quieran colaborar.

El más sofisticado de estos productos, *Citrix*, tiene su propio protocolo de línea, "ICA", que es extremadamente bueno en especial con conexiones de bajo ancho de banda. Puede ejecutar una "granja" de servidores con equilibrio de carga y tiene otras útiles facilidades. Hay clientes ICA libres que funcionan con GNU/Linux.

Todos estos productos dependen de software propietario de fuente cerrada y *Citrix* en particular resulta caro. Requiere una licencia de servidor *Windows*, una licencia *Citrix* y una licencia *Windows Terminal Server* si se usa un cliente que no sea *Windows*. Además, se necesitarán Licencias de Acceso a Clientes para cada escritorio que use el software. La licencia *Citrix* se basa en usuarios simultáneos, así es que este planteamiento puede resultar eficaz en cuanto al coste si hay muchos usuarios que necesitan acceder a una aplicación, pero no si el número de usuarios de acceso simultáneo es escaso. Hay estudios de casos documentados en <http://www.citrix.com/press/news/profiles/> que muestran que el ahorro por tener escritorios desechables de clientes ligeros es suficiente para justificar el trasladar las aplicaciones a un servidor. *Citrix* también tiene productos que permiten que las aplicaciones Unix sean transportadas de la misma manera por ICA y visualizadas en un escritorio de cliente ligero.

*Windows Terminal Server* procura una funcionalidad similar a *Citrix* excepto que usa su propio protocolo, el RDP. El cliente GNU/Linux para RDP, *Rdesktop*, es bueno pero todavía hay quien lo considera como código beta. Antes el RDP era poco

eficaz en comparación con el ICA, pero la diferencia ahora es pequeña, por no decir desdeñable. *Citrix* tiene ciertas características como el equilibrio de carga que la convierte en la mejor elección para instalaciones a gran escala cuando se puede justificar el coste extra.

Tanto *Citrix* como *Windows Terminal Server* pueden introducir latencia en la aplicación si los servidores no tienen el tamaño adecuado y la red no es lo bastante rápida.

*Tarantella* (<http://www.tarantella.com/>) se asienta en un servidor entre el escritorio y los servidores de aplicaciones. Agrega el rendimiento de *Citrix* a *Windows* y otras aplicaciones que funcionan en macroordenadores IBM y Unix, y envía el resultado a un navegador en el escritorio. Utiliza su propio protocolo de línea propietario, AIP, que parece razonable con anchos de banda bajos. Sin embargo, aumenta la latencia porque se asienta entre el usuario y la aplicación y así hace más lenta la conexión entre ambos.

Como se ha dicho anteriormente, CodeWeavers fabrica ahora una versión para servidor de su producto *CrossOver Office*. Éste funciona estableciendo una conexión segura entre el cliente y el servidor central y manteniendo una sesión X. Esto significa que la comunicación con el servidor central está cifrada y comprimida pero también necesita bastante ancho de banda para soportarlo tal y como está basado en X. No se han hecho pruebas de los requisitos de ancho de banda pero parecen ser superiores a los del ICA (*Citrix*) o AIP (*Tarantella*).

VNC es un producto OSS creado por AT&T, diseñado para visualizar una sesión de usuario que se está ejecutando en otra máquina. Consiste en un servidor y cliente, ambos disponibles para *Windows*, Unix y GNU/Linux. VNC permite que las aplicaciones se ejecuten en un entorno y que la visualización se haga en otro. Utiliza su propio protocolo, el RFB, con TCP/IP, que no es tan eficaz como ICA (*Citrix*) o AIP (*Tarantella*) y por eso necesita anchos de banda de red altos (del orden de 100Mb/s) para funcionar bien. Lamentablemente, el servidor *Windows VNC* tampoco es tan eficaz como la versión Unix y puede necesitar más potencia de procesamiento de lo que se podrían esperar. VNC puede ser muy útil como uso ocasional en la administración de sistemas, permitiendo que el control de un escritorio lo tome una persona central. En tales circunstancias, podría aceptarse una elevada latencia.

#### **13.4. Software que funcionará con un emulador**

Si nada de lo arriba dicho nos proporciona un medio para ejecutar la aplicación o un sustituto, entonces se puede ejecutar originariamente pero con su entorno operativo normal emulado en lo más alto de un sistema operativo OSS. En [http://www.linuxmednews.com/linuxmednews/967526746/index\\_html](http://www.linuxmednews.com/linuxmednews/967526746/index_html) se tratan muy bien los temas relativos a este enfoque. Todas estas técnicas tienen que ver con las licencias porque pueden implicar la ejecución de múltiples copias de la aplicación propietaria y el sistema operativo.

Esta sección tiene que ver sobre todo con las aplicaciones de *Windows* pero como las técnicas pueden aplicarse a otros escenarios se tratarán más a fondo en el Capítulo 14.

Existen dos tipos de emulación:

##### **13.4.1. Emulación del hardware**

Productos como *VMware* y *Win4lin* proporcionan emulación de hardware. Permiten que un sistema operativo normal de PC funcione como aplicación a nivel

de usuario imitando el hardware de PC Intel en las interfaces de software y sus aplicaciones para su ejecución en lo más alto de una plataforma OSS.

*VMware* no es estrictamente un emulador, pues permite que la mayoría de las instrucciones pasen directamente a través del procesador, lo que significa que sólo funcionará en una máquina de arquitectura x86. Es la oferta más completa pero es propietaria y puede consumir muchos recursos de la máquina.

*Win4lin* es similar a *VMware* y también es un producto propietario, pero menos caro. Puede ser una buena solución en casos sencillos, por ejemplo en la ejecución de aplicaciones de oficina. Es un componente del producto *Lindows* que se vende con hardware de bajo coste a usuarios locales. (Como aparentemente no usa cuentas de usuarios no privilegiadas para mantener la seguridad, *Lindows* no se recomendaría para la Administración sin antes tener en cuenta cuidadosamente la cuestión de la seguridad.)

Como el planteamiento de la emulación del hardware requiere licencias completas para el sistema operativo propietario y la aplicación, junto con el coste del emulador, se vería como una forma de ejecutar un pequeño número de aplicaciones heredadas difíciles de transferir.

Hay productos de servidores *VMware* y *Win4lin* que pueden reducir los costes de las licencias si el software propietario permite la *concesión de licencias a usuarios potenciales* y no a *usuarios coincidentes*.

Existen aplicaciones de OSS que pueden emular totalmente un entorno IA-32, por ejemplo *Bochs*, pero es probable que aún no estén listas para ser usadas en la administración.

#### **13.4.2. Emulación del software**

La emulación del software permite que los programas escritos para un entorno propietario se ejecuten directamente en el sistema operativo OSS. Toda llamada al sistema hecha por ellos se mapea (asigna) en la interfaz de sistema OSS equivalente. Lo que significa que ya no es necesario tener una copia de un sistema operativo propietario.

*Wine* permite que las aplicaciones escritas para *Windows* se ejecuten en GNU/Linux proporcionando emulación de software. *Wine* se describe más a fondo en el anexo B. El principal problema que *Wine* tiene que resolver es el gran número de llamadas al sistema *Windows* (errores incluidos) que ha de soportar.

El código *Wine* OSS se puede sacar de <http://www.winehq.org/> o de CodeWeavers en <http://www.codeweavers.com/technology/wine/download.php>.

CodeWeavers fabrica dos productos propietarios, *CrossOver Office* y *CrossOver Plugin*, que se basan en *Wine* y están pensados para soportar aplicaciones de *Windows* concretas. Aunque los productos son propietarios, periódicamente se facilitan modificaciones del código a la versión OSS de *Wine*.

*CrossOver Office* se creó para que aplicaciones como *Office* y *Notes* de Lotus funcionen originariamente en GNU/Linux. Hay algunos temas pendientes pero el producto está en pleno desarrollo. No obstante, este planteamiento puede ser adecuado para ciertos usuarios, dependiendo de sus necesidades. Ahora *CrossOver Office* también se encuentra como producto para servidores, lo que significa que no necesita estar plenamente instalado en el escritorio y puede prestar una funcionalidad similar a *Citrix*.

*CrossOver Plugin* está pensado para permitir que los accesorios del navegador que normalmente sólo se ejecutan en *Windows* lo hagan también en *Netscape*, *Mozilla* y *Galeon* en GNU/Linux. Este producto lleva más tiempo en actividad que *CrossOver Office* y funciona muy bien.

Usar estas técnicas elimina el coste de la licencia del sistema operativo *Windows* pero no la licencia de las aplicaciones. La licencia de las aplicaciones ha de ser examinada para asegurarse de que no prohíbe la ejecución de la aplicación sin *Windows*. Esta restricción se usa en algunas nuevas aplicaciones de Microsoft como táctica de bloqueo, aunque su puesta en práctica legal es cuestionable.

### 13.5. Software que se puede recompilar en OSS

En el caso de las aplicaciones escritas localmente o en nombre de la Administración y para las que se dispone del código fuente, se puede hacer que el software se ejecute en una plataforma OSS. En general, el problema de la migración del código fuente en cualquier lenguaje no es la compilación sino el uso que hace el código de las librerías del sistema, incluidas las del entorno gráfico y del sistema operativo. Esto puede significar una gran intervención manual para transferir el código. Además, toda presunción sobre el entorno subyacente, como el dar nombre a los archivos, hará que sea necesario cambiar el código fuente o reproducir el entorno, con independencia del lenguaje usado.

1. **Java.** Si se ha escrito el software Java según las especificaciones Java, entonces el programa debería funcionar sin problemas. No obstante, si se han utilizado ampliaciones propietarias, el código habrá de cambiarse y en su lugar usar módulos estándar.
2. **Visual Basic.** Un producto propietario denominado *DeLux* (<http://www.deluxsoftware.com/>) se puede usar para convertir el código de *Visual Basic* a *Kylix* (véase el punto 4 a continuación) y se pueda ejecutar originariamente en GNU/Linux. **netproject** no ha podido testar este producto. Las herramientas creadas por Microsoft pueden convertir el código de *Visual Basic* en **.NET** y elaborar un código **CIL**. El proyecto Mono OSS permite que este código se ejecute en GNU/Linux. Mono se está desarrollando con gran rapidez y una determinada aplicación puede o no trabajar dependiendo del modo en que interactúa con librerías tales como las de visualización de pantalla.
3. **C#.** Funciona cada vez más con GNU/Linux, y Ximian ha creado un compilador como parte del proyecto Mono, añadiendo vinculaciones C# a componentes fundamentales del escritorio Gnome. El proyecto Mono cuenta con un intérprete que permite que el código CIL creado por herramientas de elaboración propietarias se ejecute sin cambios en GNU/Linux. El proyecto Mono y el uso del marco de desarrollo **.NET** constituyen un área de OSS muy viva en estos momentos y la situación cambia con gran rapidez.
4. **Pascal y Delphi.** Pascal como lenguaje libre se usa poco en la actualidad, pero es el componente básico de codificación de la herramienta en rápido desarrollo *Delphi* de

Borland. Borland tiene un lenguaje nativo sobre GNU/Linux que es equivalente a *Delphi* y figura con el nombre de *Kylix*. *Kylix 2* y *Delphi 6* se han definido para usar una sintaxis de códigos compatible y tienen los entornos de soporte idénticos.

5. **C y C++.** Los programas escritos para los estándares ANSI deberían recompilarse y ejecutarse en tanto en cuanto las librerías de sistemas subyacentes sean compatibles. Por ejemplo, los programas escritos especialmente para *Windows* en general no se compilarán ni funcionarán bien con GNU/Linux debido al conjunto de llamadas tan diferentes a los sistemas operativos y las librerías en tiempo de ejecución como los del sistema de ventanas. Esta falta de conjunción suele solventarse compilando el código con *Wine lib*, que forma parte del proyecto *Wine*.

## 14. Escenario 1 - Windows

La administración tiene uno o más grupos de trabajo *Windows* interconectados, los dominios *Windows NT* PDC/BDC o *Windows 2000* Active Directory. Todos los usuarios tienen escritorios *Windows*. Todas las aplicaciones centrales se ejecutan en servidores *Windows*.

A lo largo de este capítulo la palabra *Windows* se entenderá como una versión de *Microsoft Windows*. Cuando la versión concreta sea importante, se indicará. Los ejemplos de códigos se basan en un sistema *Red Hat Linux*; otras distribuciones pueden tener sutiles diferencias.

El contenido de este Escenario debería leerse junto con los comentarios generales hechos en los capítulos anteriores.

### 14.1. Planificar la migración

Para recapitular lo que se dijo en el Capítulo 5, planificar la fase de transición es muy importante, ya que el éxito de un proyecto OSS será juzgado en gran medida por la suavidad de la transición y la calidad del servicio final. Es probable que cualquier transición práctica de un sistema a otro tenga lugar a lo largo de varios meses o incluso años. En este tiempo, los datos deben ser trasladados, el personal formado, el software instalado y la actividad de las Administraciones se desempeñará sin interrupciones.

### 14.2. Dominios

Este Escenario se puede dividir en los siguientes puntos:

#### 14.2.1. Modelo de "grupo de trabajo" Windows

Un grupo de ordenadores *Windows* que cooperan libremente en la red al haberse declarado parte del mismo "grupo de trabajo". No hay aspectos de seguridad en relación a los grupos de trabajo: sólo sirven a máquinas de grupo en listas de navegadores.

Los usuarios que quieran compartir archivos con otros pueden hacer que estén disponibles particiones es decir, partes de su jerarquía de directorio, bien para un acceso general bien a través de una contraseña.

En este modelo no existe coordinación de los nombres de usuarios y las contraseñas. De hecho, en algunas versiones de *Windows* no existe el concepto real de propiedad del archivo.

La transferencia desde un esquema de trabajo de grupo a otro implicará la recogida manual de archivos importantes, una máquina cada vez.

#### 14.2.2. Dominio NT de Windows

En este modelo, uno o más ordenadores actúan como controladores de dominios para coordinar los nombres de los usuarios y las contraseñas. Una de estas máquinas servidor es designada Controlador Principal de Dominios o PDC, y todos los cambios los maneja esa máquina. También puede haber uno o más Controladores de Dominios de Reserva o BDC para proporcionar redundancia y load-sharing.

Los dominios *Windows NT* suelen incluir uno o más servidores de archivos (que pueden ser las mismas máquinas que están ejecutando las funciones PDC y BDC). Los servidores de archivos procuran almacenamiento a los perfiles ambulantes (escritorios de usuarios, documentos y configuraciones) y también pueden proporcionar espacio para "directorios locales", almacén de archivos compartido y servicios de cola de impresión.

En un dominio bien gestionado, a los usuarios normalmente se les dice que mantengan todos los archivos en su escritorio o directorio local para que ningún dato importante esté en PCs individuales. Transferir datos de ese entorno bien gestionado a nuevos sistemas resulta relativamente sencillo, ya que los administradores de sistemas saben dónde encontrar todos los archivos que importan.

#### **14.2.3. Dominio de Active Directory en Windows 2000**

El modelo de dominio *Windows NT* se vuelve muy difícil de gestionar eficazmente para un número elevado de usuarios, por ello *Windows 2000* introdujo un modelo de dominio jerárquico. Se conoce como Active Directory o AD (Directorio Activo) y utiliza ideas del Sistema de Nombres Dominio en Internet (DNS) y del Lightweight Directory Access Protocol (LDAP, o Protocolo Ligero de Acceso a Directorios).

Como en los dominios de *Windows NT*, el AD normalmente permite a los servidores de archivos que tengan perfiles ambulantes y directorios locales de modo que sea sencillo encontrar a los archivos importantes cuando se está planeando el proceso de migración.

Dado que AD permite el acceso a LDAP, hay más opciones de migración disponibles para un sitio que use AD. Por ejemplo, debería ser posible usar los servidores de AD para conservar los datos de nombre de usuario y de la contraseña para usuarios y clientes de OSS: esto podría ser conveniente cuando una pequeña parte de la base total de usuarios ha de transferirse a OSS, ya que el proceso de gestión de usuarios puede quedar casi sin cambios.

### **14.3. Vista general de las posibles rutas de migración**

Las dos principales rutas aquí consideradas son:

1. Añadir máquinas OSS a los dominios *Windows* existentes e ir trasladando gradualmente los datos y los usuarios, y luego eliminar a los antiguos servidores propietarios. Es posible transferir a clientes y servidores independientemente.

Añadir servidores al dominio *Windows* es uno de los modos más rápidos de sacar provecho del OSS. Por ejemplo, la combinación del Sistema Operativo GNU/Linux con *Samba* da un servidor de impresión/archivos potente y de bajo coste que se puede usar en lugar de un sistema *Windows* sin gastos para el entorno de cliente existente.

Ejecutar clientes OSS en un dominio *Windows* es una forma de coexistencia de escaso riesgo, ya que no es necesario cambio alguno en relación a los servidores. Se puede usar donde un pequeño número de personas van a usar escritorios OSS en un entorno sólo de *Windows*.

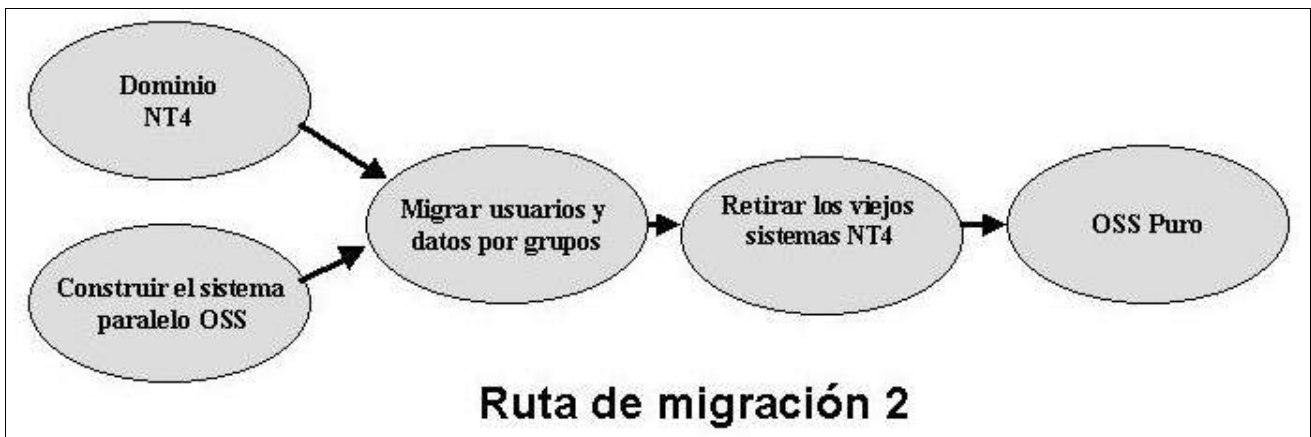
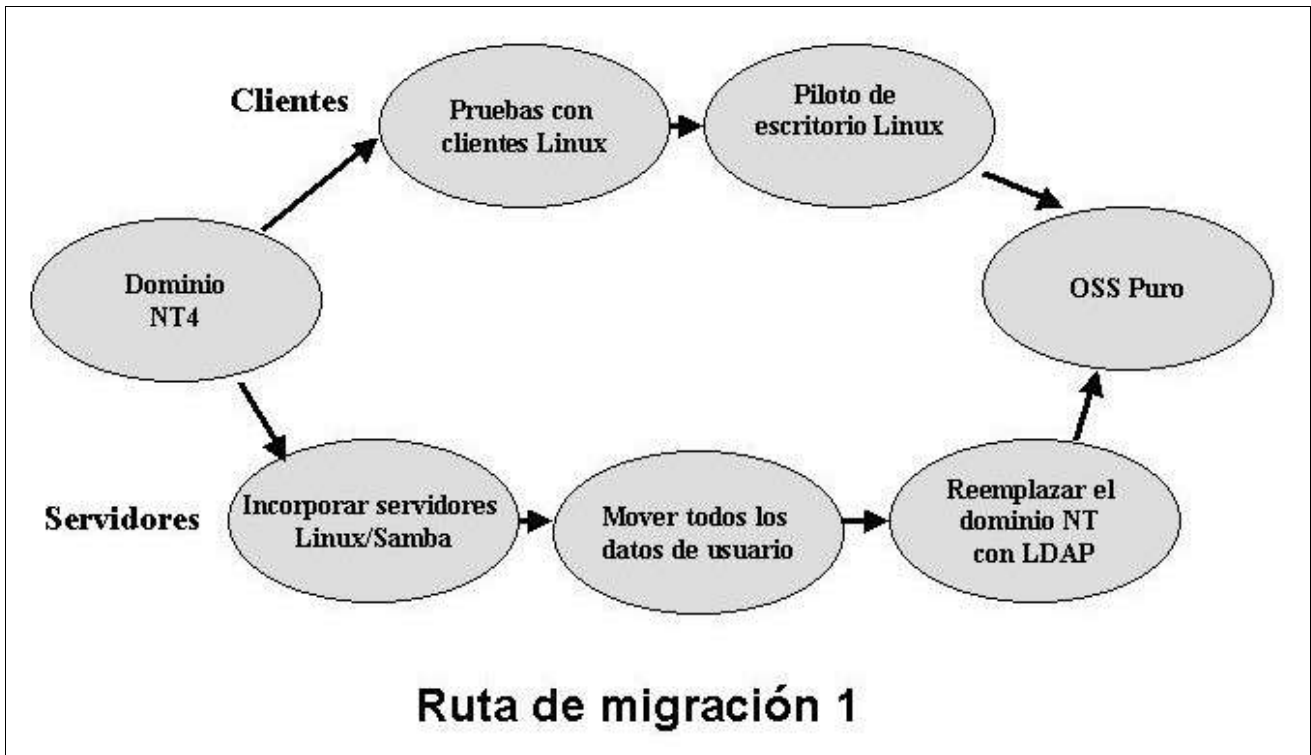
2. Construir una infraestructura paralela de tipo OSS y transferir a los usuarios y sus datos en grupos, con mínima interacción entre el sistema antiguo y el nuevo.

Es mucho más sencillo que ejecutar un sistema mixto OSS/*Windows*, pero crea una cooperación entre la gente que usa *Windows* y la que usa sistemas OSS más difícil.

Las dos rutas se resumen en los siguientes diagramas. La primera ruta proporciona una integración más firme entre el sistema antiguo y el nuevo durante la transición, pero exige bastante más planificación y trabajo de puesta en marcha.

Una limitación a la elección de ruta será el modo en que la administración está organizada y cómo se establece la correspondencia con la estructura lógica y física de la instalación informática.





Las primeras etapas de la mayoría de las rutas de migración incluyen una fase de coexistencia, donde tanto *Windows* como los sistemas OSS están en uso y con frecuencia acceden a los mismos datos. Estos modelos pueden ser muy útiles cuando se planea una migración parcial, con algunos grupos trasladándose a OSS y otros permaneciendo con el antiguo sistema.

Los pormenores técnicos de cómo hacer esos cambios se dan en la sección 14.6 más adelante. Pero primero vamos a hablar de los antecedentes técnicos y de las herramientas necesarias.

#### 14.4. Temas generales

Hay muchas similitudes entre los sistemas propietarios actuales y los sistemas OSS que podrían elegirse para sustituirlos. En particular, las interfaces de usuarios gráficas (GUI) han tendido a converger en un aspecto visual y operacional bastante estándar que reduce los problemas de cara a los usuarios finales que se pasan de un sistema a otro. Seguirá

siendo necesaria la formación de los usuarios finales, para ayudar a todo el personal que debe enfrentarse a cosas diferentes y sacar lo mejor del nuevo sistema.

Detrás de la apariencia similar de las GUI hay algunas diferencias importantes entre *Windows* y los sistemas OSS. Son especialmente visibles a nivel de la administración del sistema. Es aquí donde van a ser necesarias una mayor planificación y formación. Los sistemas OSS como GNU/Linux tienen GUIs de gestión, pero las grandes instalaciones se suelen gestionar mediante herramientas de línea de comandos, ya que se prestan para *scripting*, automatización de procesos, gestión remota y control avanzado. Es esta capacidad para automatizar las tareas lo que hace tan productivos a los administradores de los sistemas OSS y Unix.

Además de las diferencias en los procesos de gestión, hay otras igualmente importantes en el servicio prestado. Se deben haber planeado y se tratarán durante la transición.

#### **14.4.1. Nombre de usuario y palabras de paso**

Los usuarios se identifican mediante nombre de usuario y palabra de paso. En algunas Administraciones puede que usen tarjetas inteligentes o dispositivos criptográficos para reforzar el procedimiento de identificación.

##### **14.4.1.1. Nombres de usuario**

Algunas Administraciones pueden tener nombres de usuarios "estructurados" que codifican la información sobre el usuario. Por ejemplo, el nombre de usuario *cfg27* podría pertenecer a la 27ª persona registrada en el Corporate Finance Group. Otras permiten que la gente elija sus propios nombres de usuario, o simplemente usan su nombre real. Los esquemas de nombres de usuarios estructurados se pueden usar normalmente en los sistemas OSS sin ningún cambio. Los nombres de usuarios OSS no pueden empezar con un numérico, lo que causaría problemas con los nombres de usuarios estructurados donde la estructura inicial bien podría ser numérica.

Hay algunos temas que podrían afectar a los sistemas más ad-hoc. Los nombres de usuarios en los sistemas *Windows* suelen ser de los que preservan la letra y de los que no distinguen mayúsculas de minúsculas. Esto significa que si a una persona se le da el nombre de usuario de "Mary", ella puede teclear "mary", "MARY" o incluso "mArY" en el momento de la conexión sin problemas. También significa que siempre que el sistema visualice un nombre de usuario (como el del propietario de un archivo) usará la forma tecleada en principio por el administrador cuando se creó el nombre de usuario, en este caso, "Mary".

Por otra parte, los nombres de usuario en Unix y OSS son de los que distinguen mayúsculas de minúsculas. El usuario debe teclear su nombre de usuario exactamente en la misma forma en que se registró originariamente. Como convención, los nombres de usuario están compuestos en su totalidad de minúsculas y números y no se usa ningún otro carácter, siendo su longitud total de ocho caracteres.

Estas restricciones se han ido relajando mucho en los últimos años, y los sistemas modernos permitirán nombres de usuarios bastante más largos con un conjunto de caracteres más amplio. Ciertos esquemas de autenticación y autorización ahora implantan nombres de usuario que no distinguen mayúsculas de minúsculas: el esquema basado en LDAP propuesto en este documento es de ese tipo, de modo que los nombres de usuario como "Mary" y "FinancialController" son bastante posibles. Aunque hay que tener cuidado, ya que pueden haber circulando paquetes antiguos

que hagan presunciones basadas en las normas primitivas. En particular, sería poco inteligente permitir espacios u otros caracteres de puntuación en los nombres de usuario.

Una buena práctica sería la de limitar los nombres de usuario a usar los caracteres permitidos en los nombres de correo para que los nombres de usuario se puedan usar también como direcciones de correo.

#### **14.4.1.2. Contraseñas**

Los modernos sistemas OSS permiten contraseñas de casi cualquier longitud, y admiten el uso de un conjunto muy amplio de caracteres. Es una buena práctica animar al uso de contraseñas largas (10 o más caracteres) con una buena mezcla de letras, números, puntuación y mayúsculas y minúsculas. Las utilidades para la configuración de contraseñas suelen rechazar las contraseñas muy débiles a menos que se vean forzadas por un administrador, y muchos sitios pueden decidir la adopción de normas aún más exigentes.

Algunas variantes comerciales de Unix aún cortan las contraseñas en 8 caracteres por lo que, si se plantea un entorno mixto, esto se debería tener en cuenta.

La migración de contraseñas desde los sistemas propietarios existentes a los nuevos sistemas OSS no siempre es posible, ya que las contraseñas suelen mantenerse en forma resumida criptográfica. Puede que haya que incluir en el plan de transición la reemisión de nuevas contraseñas para todos los usuarios, o tal vez una fase de recogida y sincronización de contraseñas.

#### **14.4.2. Servicios de autenticación**

Toda red de más de unos cuantos ordenadores necesita un servicio de autenticación y designación de nombres en red. En *Windows NT* esto se conoce como Controlador de Dominios. En sistemas *Windows* posteriores es el Directorio Activo. También se ha instalado mucho el NDS de Novell, y otros sistemas propietarios tienen sus propios sistemas de autenticación y designación de nombres.

La mayoría de los sistemas OSS y Unix pueden interactuar con la mayoría de los servicios de autenticación y designación de nombres. GNU/Linux es especialmente fuerte en este sentido. El servicio propuesto en este documento se basa en LDAP, pero también se pueden usar servicios de autenticación y designación de nombres múltiples, que pueden ser útiles en la fase de transición.

#### **14.4.3. Archivos**

Una parte muy importante en todo plan de transición se refiere a la migración de datos desde el sistema antiguo al nuevo. Si se ha planeado una migración tipo "big bang" entonces se tratará de una operación única, pero si como es más probable lo que se pretende es una ejecución paralela, será necesario un acceso a los archivos en multiplataforma. Se debe tener mucho cuidado en evitar la pérdida de datos y la confusión que podría resultar del hecho de tener copias separadas modificables de un archivo tanto en el entorno viejo como en el nuevo.

##### **14.4.3.1. Formato y contenido**

Este es el tema más obvio de toda migración, y de él se habla más a fondo en el punto 14.8 más adelante. El planteamiento normal consiste en usar las aplicaciones OSS que pueden leer archivos escritos por la aplicación propietaria a la que sustituyen, aunque en algunos casos puede ser más apropiado planear una conversión de formato en masa como parte del proceso de migración.

Los datos especiales como macros y *scripts* pueden necesitar la atención de programadores especializados durante la migración.

#### 14.4.3.2. Nombres de archivos

Como en el caso de los nombres de usuarios, los nombres de archivo *Windows* no distinguen mayúsculas de minúsculas y (en cierta medida) preservan la letra. Algunas aplicaciones *Windows* tienden a usar mayúsculas en la primera letra de los nombres de archivo y hacen otros cambios de los que el usuario puede o no ser consciente. El entorno *Windows* también lleva la herencia del formato de nombres de archivo del DOS "8.3", que todavía aparece en algunas utilidades. Los nombres de archivo *Windows* suelen contener espacios, y normalmente usan el conjunto de caracteres Unicode. *Windows* usa "\" como separador de directorios.

Aunque es menos obvio para los usuarios de GUI, los nombres de archivos totalmente *Windows* deben incluir una "letra de unidad" que indique el dispositivo físico que guarda el archivo, o bien deben tener el nombre real del servidor si el archivo está en una "unidad de red". Estas restricciones pueden ser un problema para los gestores de grandes sistemas *Windows* que tratan de proporcionar un servicio sin fisuras de cara a los cambios de hardware.

Otros sistemas propietarios tratan a los nombres de archivos de distintas maneras. *VMS*, por ejemplo, tiene nombres de archivos que no distinguen mayúsculas de minúsculas y que suelen incluir un punto y pueden llevar un número de versión tras un punto y coma.

En los sistemas Unix y OSS, los nombres de archivos siguen normas diferentes. Aquí, los nombres de archivos distinguen mayúsculas de minúsculas y el sistema no hace cambio alguno a los nombres facilitados por el usuario. Los nombres usan un conjunto de caracteres de 8 bits determinado por el lugar actualmente en uso (en la mayor parte de Europa, el conjunto de caracteres es ISO 8859-15). Los únicos caracteres que Linux no permite en los nombres de archivos son el separador de directorios ("/") y el carácter nulo. Aunque en la práctica no resulta inteligente incluir caracteres que no se pueden imprimir, por ejemplo, el sistema de archivos FAT32 de *Windows* no puede almacenar los primeros 32 códigos ASCII ni ninguno de éstos: ", \*, :, <, >, ?, \ o |. Los espacios en los nombres de archivo sí se permiten, aunque su presencia requiere que los usuarios de líneas de comandos sean más cuidadosos en las citas textuales.

Los sistemas OSS y Unix no usan letras de unidad, y no exigen que el nombre real del servidor de archivos sea parte del nombre de archivo absoluto cuando accede al archivo en la red. En su lugar, el sistema presenta todos los archivos como parte de una jerarquía sin fisuras. Junto con el uso de enlaces simbólicos en el sistema de archivos y los automontadores de datos controlados, esto da a los administradores del sistema gran flexibilidad en la separación del nombre absoluto de un archivo de su ubicación física de almacenamiento.

Casi todos los nombres de archivos *Windows* se pueden transferir directamente a servidores OSS sin ningún cambio. La única excepción que es probable que encontremos en la práctica es la de los nombres de archivo que contienen el carácter "/", que habrá de ser modificado durante la transición. Es probable que los usuarios de herramientas de GUI nunca adviertan que los nombres de archivos ahora distinguen mayúsculas de minúsculas ya que ellos sólo teclean esos nombres cuando crean el archivo.

#### 14.4.3.3. Acceso doble

Es probable que muchos planes de migración incluyan un período de funcionamiento paralelo en el que algunas personas todavía estén usando sistemas propietarios. Cuando es preciso que miembros de los dos grupos accedan a los archivos, puede que se necesiten disposiciones especiales para ello.

Al compartir archivos en los sistemas *Windows* se usa el protocolo SMB (Bloque de mensajes del servidor), que consiste en una tecnología muy compleja con múltiples niveles de compatibilidad hacia atrás. Se usa con servidores de archivos dedicados y también en modo "entre iguales", donde los PC individuales facilitan partes de sus sistema de archivos en la red. Es probable que los entornos de la administración bien gestionados se basen en servidores dedicados más que en la compartición ad hoc.

Los archivos de usuario no compartidos en un entorno *Windows* pueden mantenerse en varios sitios diferentes:

1. En un disco local del escritorio del usuario o PC laptop: por ejemplo uno al que se hace referencia como "la unidad C".
2. En el "perfil ambulante": que incluye las configuraciones preferidas y también el contenido del escritorio *Windows* y (normalmente) la carpeta "Mis documentos". El perfil ambulante se conserva localmente en cualquier PC que el usuario esté usando activamente, y se sincroniza a un depósito de perfiles en tiempo de salida. Esto proporciona una facilidad de copia de seguridad manual, pero puede tener serias implicaciones de rendimiento en el caso de los usuarios que informan salidas muy escasas.
3. En un "directorio local" de un servidor de archivos gestionado centralmente. Es una opción corriente para grandes redes de sistemas de escritorios ya que es fácil gestionar bien las copias de seguridad.

No es sensato tratar de proporcionar un doble acceso a archivos que están en un escritorio individual o en PCs laptop, pues los archivos de los discos locales o los perfiles ambulantes deberían trasladarse a servidores de archivos gestionados al principio del proceso de migración.

El mecanismo principal de acceso a archivos en red de los sistemas Unix y OSS es el sistema de Archivos en Red (NFS). Es un protocolo mucho más sencillo que el SMB, y sus especificaciones siempre han estado disponibles.

Las opciones para poner en marcha el doble acceso caen en dos categorías amplias: añadir soporte de protocolo doble a los servidores, o añadir soporte de protocolo doble a los clientes. Como lo demás es igual, normalmente es más fácil cambiar los servidores que los clientes, y casi siempre es más fácil ajustar los sistemas OSS que los propietarios. Las opciones se resumen en la siguiente tabla:

	<b>Servidores <i>Windows</i></b>	<b>Servidores OSS o Unix</b>
<b>Cientes <i>Windows</i></b>	El acceso a los archivos de SMB es estándar.	Los servidores funcionan con SMB usando el paquete Samba. es un software maduro con un excelente rendimiento.
<b>Cientes OSS</b>	<p>Los clientes de GNU/Linux pueden acceder a partes de SMB. Esto es bastante fácil cuando las máquinas de los clientes sólo tienen un usuario a la vez, pero se implica menos cuando se usan máquinas en tiempo compartido.</p> <p>Normalmente las variantes comerciales de Unix no tienen capacidad para clientes de SMB.</p> <p>Se puede añadir un servicio NFS a los servidores <i>Windows</i>, pero puede ser muy caro.</p>	<p>El acceso a los archivos de NFS es estándar.</p> <p>Los clientes de GNU/Linux pueden usar el SMB si lo desean como parte de un plan de migración, pero es menos eficaz.</p>

## 14.5. Herramientas

En este punto se habla de algunos de los componentes clave del OSS que se usarán durante la migración desde sistemas propietarios.

### 14.5.1. Samba

*Samba* es un paquete de servidor de impresión y de archivo para los sistemas OSS. Pone en marcha el protocolo SMB de Microsoft y en muchos casos sustituye completamente las funciones de un servidor *Windows*. *Samba* también puede actuar como un Controlador de Dominios de *Windows NT* y es capaz de almacenar datos de gestión de dominios en un directorio al que se accede mediante el LDAP.

*Samba* también proporciona herramientas de clientes SMB adecuadas para *scripting*, que son muy útiles al diagnosticar problemas con las redes SMB y al hacer migraciones de archivos en bloque desde los servidores *Windows*.

*Samba* es mantenido por un grupo central de unos 30 voluntarios muy activos de todo el mundo. En <http://www.samba.org/> encontrará más información al respecto.

### 14.5.2. OpenLDAP

*OpenLDAP* es una realización del Protocolo Ligero de acceso a Directorios (LDAP). Incluye un servidor de directorios, un conjunto de herramientas de gestión y de acceso a los datos y un conjunto de librerías para soportar el LDAP en otras aplicaciones.

*OpenLDAP* es mantenido por un pequeño grupo central más un gran número de colaboradores. Un miembro del grupo central trabaja en el proyecto a tiempo completo.

### 14.5.3. NSS y PAM

NSS es el Name Server Switch (Conmutador de Servidores de Nombres): una tecnología utilizada por GNU/Linux y algunas variantes comerciales de Unix para permitir que diferentes servicios de nombres sean usados al buscar los nombres de anfitrión, nombres de usuario, nombres de grupo, etc. Hay disponibles muchos módulos, de los cuales los que más tienen que ver con este proyecto son:

1. archivos: simples búsquedas basadas en los archivos de texto locales.
2. DNS: búsquedas de nombres de anfitrión basadas en el Sistema de Nombres Dominio.
3. LDAP: búsquedas basadas en el LDAP, la mayoría de nombres de usuarios y nombres de grupo pero que también se puede usar para otros muchos fines.
4. SMB: búsquedas usando el protocolo SMB de *Windows* (Véase el punto 14.5.5. más adelante).

PAM es el sistema de Pluggable Authentication Module System (Módulo de Autenticación Conectable). Como el NSS, es común a GNU/Linux y varios derivados comerciales de Unix. PAM también está disponible en *FreeBSD*. PAM permite una gran flexibilidad en la configuración del proceso de autorización y autenticación. Los módulos incluyen:

1. LDAP: usa operaciones de vinculación a LDAP para comprobar las credenciales del usuario.
2. SMB: usa operaciones de dominio *Windows NT* para comprobar las credenciales del usuario.
3. Acceso: restringe el acceso a los servicios en red.
4. Cracklib: aplica controles de calidad a las nuevas contraseñas.

### 14.5.4. Acceso a archivos SMBFS de GNU/Linux

*Samba* permite que un sistema OSS proporcione un servicio de archivos a los clientes *Windows*. SMBFS trabaja de otra forma: permite a un sistema OS acceder a archivos que están en servidores *Windows*. SMBFS cuenta con la mayoría de las distribuciones de GNU/Linux pero no se suele encontrar en los sistemas comerciales Unix.

El modelo de control de accesos utilizado por los sistemas de archivos *Windows* es diferente del usado por GNU/Linux y otros sistemas OSS, de manera que hay algunas limitaciones en cuanto a lo que se puede conseguir con SMBFS.

### 14.5.5. Winbind

Otro producto del equipo *Samba*, *Winbind*, permite a las máquinas individuales GNU/Linux adjuntarse a los dominios *Windows NT*. Mantiene una correspondencia entre los autenticadores de *Windows NT* (SIDs) y los UIDS tipo Unix y GIDs. *Winbind* puede hacer muchas otras cosas que reducen la carga en los administradores de sistemas, como es la configuración de entornos de tipo Unix para personas cuando se conectan por primera vez.

El punto débil de *Winbind* en las grandes redes es que cada ordenador de cliente construye su propia correspondencia entre los autenticadores de *Windows* y los de Unix. Esto puede causar problemas en las últimas etapas de la migración cuando se introducen los servidores de archivos OSS.

Al usar *Winbind*, los nombres de usuario y los nombres de grupo utilizados por GNU/Linux se forman mediante una concatenación del nombre de dominio *Windows NT* con el nombre de usuario *Windows NT* para formar una cadena de caracteres única. Esto puede llevar a cierta confusión, ya que muchas utilidades corrientes de tipo Unix sólo proporcionan espacio a nombres de usuario de ocho caracteres. Los nombres más largos generados por *Winbind* se cortaran en la visualización.

## 14.6. Migración del entorno de sistema operativo

### 14.6.1. Añadir servidores individuales de GNU/Linux a un Dominio Windows NT existente

La configuración es extremadamente sencilla:

1. Instalar un servidor GNU/Linux, dando una dirección fija de IP.
2. Asegurarse de que los paquetes Samba están instalados: se necesitan típicamente *samba*, *samba-common* y *samba-client*. Se suelen incluir en una instalación de “servidor”.
3. Editar `/etc/samba/smb.conf`, establecer el modo de seguridad del **dominio**, y definir el nombre de dominio. Hacer una lista de los PDC y BDC como servidores de contraseñas. Definir las partes que la máquina ha de servir.
4. Crear los directorios que han de compartirse, y establecer la propiedad y los permisos adecuados.
5. Unir la máquina al dominio de *Windows NT* existente usando la contraseña de Administración de Dominio (o cualquier otro nombre de usuario y contraseña que tenga la facultad de hacerlo):

```
Smbpasswd -j DOMAINNAME -r PDCNAME -U Administrator
```

6. Iniciar samba y disponerlo para que se reanude en reinicio:

```
/etc/init.d/smb Stara  
chkconfig smb on
```

El servidor aparecerá ahora en las listas de navegación y se puede usar como un servidor de *Windows NT*.

### 14.6.2. Ejecutar escritorios GNU/Linux en dominios Windows NT

#### 14.6.2.1. Configuración sencilla para pocas máquinas

En las primeras etapas del ensayo de las herramientas de OSS, es muy útil ejecutar las máquinas individuales de GNU/Linux con configuraciones muy sencillas, a las que se les puede dar acceso a archivos en servidores *Windows* para los tests de compatibilidad y migración usando el comando **smbmount**.

Montaje es el término de Unix/OSS para hacer que un disco o parte de un sistema de archivos remoto forme parte de la jerarquía de archivos de la máquina local. El proceso se suele hacer automáticamente en tiempo de arranque bajo el control del archivo `/etc/fstab`, pero también se puede hacer de manera interactiva. Por ejemplo, el comando para llevar un CDROM de norma ISO al sistema de archivos bajo `/mnt/cdrom` sería:

```
Mount /dev/cdrom /mnt/cdrom
```



El comando **mount** suele tener el uso restringido al superusuario por razones de seguridad. Esto no es un problema cuando la máquina la está usando un administrador de sistemas, pero puede ser difícil si está implicado un usuario no técnico. GNU/Linux proporciona varias formas de tratar este problema:

1. Usar una entrada especial en **/etc/fstab** que permita a usuarios corrientes montar ciertos objetos predefinidos. Este es el modo habitual de permitir que los disquetes y los CDROM sean montados a petición. Los archivos del dispositivo montado suelen mostrarse como propiedad de cualquiera que haya hecho que se monte el dispositivo.
2. Usar un programa **setuid-root** para hacer la operación privilegiada, habiendo comprobado antes que es segura. Este es el modo más fácil de manipular el montaje de partes remotas de *Windows*.
3. Usar un **automounter** para montar sistemas de archivos cuando se accede a ellos por primera vez y desmontarlos cuando ya no se usan. El automontador funciona como un daemon y suele ser accionado por datos de configuración de amplitud de red. Este método requiere más esfuerzo para su configuración que los otros, pero es extremadamente útil en las redes grandes.

En este esquema, usaremos los comandos **smbmount** y **smbumount** para hacer que una parte de *Windows* existente aparezca como parte del sistema de archivo GNU/Linux local. En los sistemas *Red Hat Linux* estos forman parte del paquete *samba-client*, por lo que debe asegurarse de que ha instalado los paquetes *samba-client* y *samba-common*. Estos programas están pensados para que a las piezas críticas se les pueda dar privilegios de usuario *root*, aunque no se instalan normalmente de esa forma de manera predeterminada, así es que algunos comandos se pueden ejecutar como *root* antes de que se usen por primera vez:

```
Chmod u+s /usr/bin/smbmnt /usr/bin/sbumount
```

Cabe observar que el comando cambia **smbmnt** más que **sbumount**. Esto es importante porque **smbmnt** sólo encapsula las funciones de **sbumount** que requieren privilegios de usuario *root*. Una vez hecho esto, cualquier usuario puede usar comandos **smbmount** y **sbumount** y funcionarán con los necesarios privilegios de *root*.

Ahora cualquier usuario puede hacer que una parte de SMB de *Windows* esté disponible como parte del sistema de archivos GNU/Linux montándolo en un directorio que ya pose. Cualquier archivo que ya está en el directorio será invisible mientras la parte de SMB esté montada en la parte superior.

Como ejemplo, supongamos que el usuario de GNU/Linux **fred** quiere acceder a archivos de un servidor de *Windows NT* llamado NT4SERVER en el dominio THESTATE, que son compartidos con el nombre de FREDERICK y pertenecen al usuario de *Windows* FREDERICK. **fred** comienza por hacer un nuevo directorio para montar la parte de *Windows* en:

```
mkdir ~/ntfiles
```

(La notación “~/” significa “en mi directorio directo”). Esto sólo hay que hacerlo una vez. Ahora, montar la parte remota:

```
smbmount //nt4server/frederick ~/ntfiles \  
-o username=frederick -o workgroup=thestate
```

El comando debería teclearse en una línea, o partirlo con caracteres de continuación de línea “\” como se ve aquí. Sugeriré la contraseña FREDERICK en el servidor, y luego montará la parte de *Windows* en la parte superior del directorio *ntfiles* en el directorio directo de **fred**. Para evitar teclear lo mismo en cada conexión (*login*), se puede poner en un archivo de *script* o incluso hacer que forme parte del proceso de conexión de **fred**.

La parte montada ahora se comporta casi como si fuera parte del disco local. Los archivos pueden ser creados, borrados y editados. Aunque hay algunas advertencias. En particular, no hay intento de establecer correspondencia entre el control de acceso de tipo Unix y los ACL de *Windows NT*, por lo que los comandos para cambiar la propiedad o el modo de los archivos y directorios en la parte montada no tendrá efecto alguno.

Antes de salir, debería ser inteligente y desmontar la parte:

```
smbumount ~/ntfiles
```

También esto podría convertirse en algo automático en el proceso de salida, en caso necesario.

El proceso descrito en esta sección no crea ningún vínculo permanente entre las cuentas en servidores de GNU/Linux y las existentes en servidores de *Windows NT*, por lo que los nombres de usuario y las contraseñas deben mantenerse por separado en cada máquina. El esfuerzo de gestión que conlleva puede resultar excesivo en poco tiempo ya que el número de máquinas crece, y este esquema realmente sólo es adecuado para pequeños entornos de prueba.

#### 14.6.2.2. Configuración elaborada para despliegues mayores

Cuando se necesita un mayor despliegue piloto del escritorio OSS, puede seguir siendo conveniente mantener los servicios de autenticación y archivos en los servidores de *Windows NT* existentes. El daemon *winbind* de *Samba* proporciona una forma fácil de vincular los dos entornos.

*Samba* y *Winbind* son piezas estándar de la distribución *Red Hat Linux*, pero no se pueden instalar de forma predeterminada en las configuraciones del escritorio. Para usar *Winbind* se deben instalar los siguientes paquetes; *samba*, *samba-common* y *samba-client*.

El archivo **/etc/samba/smb.conf** debe editarse para mostrar el nombre correcto de dominio *Windows NT* en la línea del **grupo de trabajo**, y poner el sistema en modo de seguridad **dominio**. Los datos de la configuración *Winbind* también van en la sección global de este archivo, por ejemplo:

```
# separate domain and username with '+', like
DOMAIN+username
winbind separator = +

# use uids from 10000 to 20000 for domain users
winbind uid = 10000 - 20000

# use gids from 10000 to 20000 for domain groups
winbind gid = 10000 - 20000

# allow enumeration of winbind users and groups
winbind enum users = yes
winbind enum groups = yes
```

```
# give winbind users a home directory location
template homedir = /home/winnt/%D/%U

# and a shell
template shell = /bin/bash
```

Para que *Winbind* funcione, hay que ejecutar ciertos servicios. Para iniciarlos y asegurarse de que se inician en cada arranque, los comandos son:

```
chlconfig smb on
chkconfig winbind on
/etc/init.d/smb start
/etc/init.d/winbind start
```

Ahora hay que unir la máquina al dominio *Windows NT*. Esto exige un número de usuario y una contraseña de *Windows NT* con los permisos adecuados (normalmente Administrator):

```
smbpasswd -j DOMAINNAME -r PDCNAME -U Administrator
```

Ahora debería ser posible obtener listas de los usuarios y grupos *Windows* con el comando `wbinfo`:

```
wbinfo -u
wbinfo -g
```

Para hacer que el sistema pueda disponer de los datos de *Winbind*, es necesario editar los archivos de configuración PAM y NSS. Esto debería hacerse con gran cuidado, ya que es posible encontrarse con que no puede acceder al sistema si esos archivos están dañados. En `/etc/nsswitch.conf` añadir la palabra **winbind** a las líneas **group** y **passwd**. En `/etc/pam.d/system-auth` añadir una línea como la siguiente:

```
auth sufficient /lib/security/pam_winbind.so
use_first_pass
```

justo después de la línea **auth** equivalente que usa **pam\_unix**, y una como la siguiente:

```
password sufficient /lib/security/pam_winbind.so
use_first_pass
```

justo después de la línea **password** equivalente que usa **pam\_unix**.

En esta etapa será necesario reiniciar el Daemon Cache de Servicio de Nombres:

```
/etc/INIT.d/nscd restar
```

La traducción de los grupos y normas de usuarios de *Windows* a formato de archivo de contraseña tipo Unix ahora se puede ver con:

```
getent passwd
getent group
```

Para automatizar la creación de directorios directos de usuario en la primera conexión, añada esta línea a la parte **session** de `/etc/pam.d/system-auth`:

```
session required /lib/security/pam_mkhome.so
skel=/etc/skel/umask=0022
```

(Asegúrese de que lo arriba indicado se introduce como una única línea mejor que en las dos líneas en que se presenta aquí). Observe que esto creará un directorio

Unix *home* separado para el usuario en cada puesto de trabajo que usen. También puede ser útil el poner un *script* en el directorio **/etc/skel** para que cada usuario monte automáticamente sus archivos de *Windows NT* en un lugar estándar y en tiempo de conexión.

#### 14.6.3. Ejecutar escritorios GNU/Linux en dominios de Directorio Activo

En principio, las máquinas de escritorio GNU/Linux pueden unirse a los dominios de AD (Directorio Activo) en la misma forma que se unen a los dominios de *Windows NT*. De hecho, si el dominio AD está funcionando en modo de compatibilidad NT, se puede usar exactamente el mismo proceso.

Los dominios de AD también brindan la posibilidad de usar el LDAP para autenticación y búsqueda de datos. Es el mismo esquema que se propone para redes mayores de sistemas OSS puros, y vale la pena tenerlo en cuenta. Al ampliar el esquema de AD para incluir datos Unix, se podría gestionar a los usuarios de escritorios y usuarios OSS con herramientas de administración AD. Es preferible almacenar los datos centralmente al esquema *Winbind* utilizado con *Windows NT*, ya que mantiene la correspondencia entre los ID de Unix y los de *Windows NT* en la debida forma en todas las máquinas.

#### 14.6.4. Cambiar PDC/BDC Windows NT por Samba+LDAP

*Samba* puede asumir el papel del Controlador Principal de Dominios, lo que permite eliminar a todos los servidores *Windows* incluso si algunos clientes *Windows* aún son requeridos. Nótese que no es posible sustituir precisamente el PDC o un BDC de un dominio: todos los controladores de dominios deben estar funcionando en el mismo sistema, *Windows* o *Samba*. Esto es en parte así porque el protocolo de replicado de PDC-BDC no ha sido construido a la inversa. Además, los Controladores de Dominios adoptan un planteamiento diferente de cara a la capacidad de recuperación; la delegan a los servidores LDAP donde los datos están almacenados en realidad.

Configurar un Controlador de Dominios LDAP+Samba es una tarea demasiado amplia para describirla en detalle aquí, pero una persona con experiencia puede realizarla en un día más o menos. La tarea más amplia es la de planificar la migración de los nombres de grupo y los nombres de usuarios desde un dominio existente. Parte del trabajo se cubre en el *Samba-LDAP-HOWTO* de IDEALX (véase la referencia que se hace en el punto 14.12 más adelante) La misma fuente proporciona un conjunto de esqueletos de las herramientas de migración que pueden ser una base muy buena para construir sobre ella.

En resumen, el proceso es así:

1. Instalar un servidor (o servidores) OSS con *Samba* y *OpenLDAP*. Puede ser necesario construir *Samba* desde la fuente; *Red Hat Linux 7.3*, por ejemplo, no incluye la versión admitida por el LDAP.
2. Añadir las definiciones del esquema *Samba* al servidor LDAP.
3. Configurar el servidor LDAP con una estructura de árbol, básica y adecuada, de directorio y Nombre Distinguido (DN), en lo posible usando las herramientas de IDEALX para poblar el árbol con entradas “boilerplate”.
4. Iniciar *Samba* y testar la función de Controlador de Dominios.

5. Usar **pwdump** en el PDC para hacer una lista de todas las entradas de usuarios en el SAM. Transferir el resultado al servidor OSS como archivo de texto.
6. Configurar la herramienta **smbldap-migrate-accounts.pl** de IDEALX para establecer correspondencia con el entorno en construcción. Esto no es nada trivial ya que hay muchas opciones a tener en cuenta.
7. Ejecutar **smbldap-migrate-accounts.pl** en los datos transferidos desde el PDC. Esto creará entradas en LADP para todos los usuarios de dominios. También establecerá sus contraseñas SMB para concordar las usadas con *Windows NT* (pero ello no habilitará las conexiones de Unix o GNU/Linux, ya que las contraseñas de *Windows NT* están aleatorizadas y para los sistemas OSS se usa otro esquema hash).

La herramienta puede crear directorios *home* si se desea.

8. Copiar ficheros de usuario y ambulantes desde los servidores *Windows* a los nuevos servidores OSS, o reconectar los servidores *Windows* existentes al dominio servido por los Controladores de Dominio de *Samba*.

Las redes grandes pueden necesitar servidores múltiples de LDAP con réplica de datos para su resiliencia. Si un Controlador de Dominios *Samba* tiene que ver con cada servidor LDAP, se puede realizar un esquema muy parecido a la configuración PDC/BDC de *Windows*.

Hay otros muchos temas a tener en cuenta, a saber:

1. La elección de herramientas para la gestión de usuarios.
2. Cómo se van a establecer las correspondencias entre los grupos de *Windows NT* y las ACL con los grupos de tipo Unix y sus ACL.
3. Si se va a usar o no un nuevo nombre de dominio para el servicio basado en el OSS.
4. Cómo crear aleatorizaciones de contraseñas que puedan ser usadas por el sistema OSS (o si continuar usando aleatorizaciones *LANMAN* o *Windows NT*, incluso en un entorno OSS puro).

#### **14.6.5. Cambiar el Directorio Activo Windows 2000 por el LDAP**

El grueso de los datos que hay en un Directorio Activo está en el almacén al que se accede mediante el LADP. A primera vista, esto facilitaría la sustitución de los servidores de AD por sus equivalentes OSS. Lamentablemente, este no es el caso: los sistemas *Windows 2000* no usan LDAP puro para el acceso a todos los datos, y usan una variante no estándar de Kerberos para la autenticación.

Varios equipos de OSS están trabajando para fijar el problema, pero en el momento de escribir este informe el único modo factible de soportar los clientes *Windows 2000* y *Windows NT* es ejecutarlos en dominios *Windows NT* tal y como se describe más arriba.

### 14.6.6. Ejecutar Infraestructura GNU/Linux paralela y transferir usuarios en grupos

#### 14.6.6.1. Reemplazar todos los clientes de Windows por GNU/Linux

Este es el más sencillo de todos los posibles esquemas de migración. La interacción entre los sistemas OSS y *Windows* se limita a la transferencia de una vez de los archivos de usuarios. En líneas generales, el proceso es como sigue:

1. Construir el entorno central OSS, incluyendo a los servidores LDAP para albergar los datos de configuración y nombres de usuarios, los servidores de instalación master, uno o más servidores de archivos y de impresión y suficientes puestos de trabajo de clientes para el personal que gestiona los sistemas.
2. Implantar una unidad de formación y desarrollo, con suficientes puestos de trabajo en escritorio para la formación de grupos de personas del tamaño adecuado. La tarea inicial de esta unidad consiste en validar y afinar la configuración del puesto de trabajo antes del despliegue principal.

En esta etapa, el proceso de construcción de puestos de trabajo debería estar ya terminado, para que las máquinas se puedan configurar con el menor esfuerzo humano posible. Es muy importante que todas las máquinas de escritorio estén instaladas *exactamente* de la misma manera durante la fase de despliegue principal, por lo que esto debe comprobarse cuidadosamente.

3. Usar la unidad de formación y desarrollo consultando a los representantes clave de la base de usuarios para generar entusiasmo por el proyecto y recoger comentarios y sugerencias sobre la interfaz de usuario. Hacer los cambios que sean necesarios para llegar a la configuración de despliegue.

Acordar las necesidades y el programa de formación.

4. Construir un conjunto de nuevos puestos de trabajo de escritorio suficientes para sustituir el equipo que está utilizando el primer grupo que ha de transferirse a los sistemas OSS.
5. Registrar el primer grupo de usuarios en el nuevo sistema.
6. Formar al primer grupo de usuarios en el nuevo sistema.
7. En caso necesario, reestablecer cualquier configuración cambiada durante la formación para que todos empiecen con un entorno conocido.
8. Sustituir los PC de escritorio del primer grupo por los sistemas OSS prefabricados.

Al mismo tiempo, copiar los archivos del grupo en los nuevos servidores de archivos y establecer la copia original como de sólo-lectura.

9. Dar soporte activo al primer grupo mientras se acostumbran a trabajar con el sistema OSS.
10. Mejorar, en la medida en que sea necesario, los PC retirados del primer grupo, e instalar la imagen de puesto de trabajo estándar.
11. Repetir desde el punto 5 con el siguiente grupo de usuarios.
12. Cuando todos los usuarios se hayan pasado a los sistemas OSS, hacer copias de todos los archivos que haya en los servidores antiguos y liberarlos.

#### 14.6.6.2. Conservar algunos clientes de Windows

Cuando haya que retener a algunos clientes de *Windows* (por ejemplo para soportar funciones cuya transferencia no resulte adecuada por su coste debido a que el software no es portátil) hay dos opciones principales:

1. Retener un pequeño dominio de *Windows* usando uno o más servidores de *Windows*.
2. Soportar los clientes de *Windows* desde servidores basados en OSS mediante *Samba*.

La ruta elegida dependerá de la razón por la cual se está reteniendo a los clientes de *Windows* y de su distribución geográfica.

En cualquier caso, es probable que se necesite recurrir a *Samba* en uno o más de los nuevos servidores, para poder compartir archivos entre los clientes de *Windows* y los basados en OSS.

### 14.7. Migración de aplicaciones del servidor

#### 14.7.1. Servidores Web: migrar de IIS a Apache

El servidor web habitual de *Windows* es *IIS* (Internet Information Server), que proporciona servicios de HTTP, FTP y Gopher en un solo paquete. *IIS* tiene una cierta reputación por problemas de seguridad y estabilidad, que ha conducido a muchas organizaciones a sustituirlo por un servidor web alternativo. De hecho, tras una serie de debilidades especialmente serias explotadas en 2001, los analistas y Gartner remitieron a sus clientes una recomendación firme de que *IIS* no fuera utilizado para servicios críticos mientras que no hubiera sido reescrito completamente por Microsoft.

Hay un conjunto de servidores web de entre los que escoger cuando se plantea la sustitución de *IIS*. Muchos de ellos son OSS o bien tienen condiciones de licenciamiento bastante abiertas. Algunos de los servidores más ampliamente utilizados se tratan en el apartado 11.4.2. Cuando se migran sitios web desde *IIS*, la opción habitual suele ser *Apache* - a menudo con módulos PHP o Perl. *Apache* funciona sobre GNU/Linux, FreeBSD, y casi todas las otras variantes de Unix y también sobre Windows. Se dispone de un amplio abanico de alternativas para la migración.

##### 14.7.1.1. Temas de migración

#### 1. Nombres de archivo y URLs

Al desplazar un sitio web sencillo de *IIS* o *Windows* a *Apache* en GNU/Linux o Unix, la cuestión principal a tener en cuenta es que el sistema de archivos *Windows* ignora el tipo de letra mayúscula/minúscula en los nombres de archivos, pero la mayoría de los sistemas de archivos Unix o GNU/Linux distinguen de mayúscula a minúscula. Como la jerarquía de las páginas web normalmente se representa directamente en el sistema de archivos, esto significa que los URL van a distinguir de mayúscula a minúscula cuando se pasan al entorno GNU/Linux o Unix. (Esto no sería una cuestión a tratarse si *Apache* fuera utilizado en un servidor de *Windows*).

Un tema menos corriente es que *IIS* parece aceptar a “\” y “/” como separadores de componentes, y debe traducir “/” a “\” para el sistema de archivos *Windows*, pero parece permitir que “\” trabaje de forma nativa. Así

pues, se podría hacer referencia a un archivo en un URL como **mydir\thisfile.jtml** y **mydir/thisfile.html**.

Ninguno de estos asuntos afectará a un sitio web bien escrito y consistente. Lamentablemente, los sitios construidos con software *Windows* suelen tener un uso no coherente de las mayúsculas y las minúsculas, y a veces tienen el carácter “\” en los URL donde la estructura de archivo en el sitio web contiene un subdirectorio. De hecho, el modelo de sitio web distribuido con las primeras versiones de *IIS* visualiza los dos asuntos. En *Apache* hay tratamientos sencillos para los dos problemas, que se demuestran en el ejemplo que damos más adelante en este capítulo. Aunque, como norma general, es mejor corregir esos problemas en los datos del sitio web.

## 2. Mapas de imagen tipo servidor

Algunos sitios web antiguos usaban en la parte del servidor una representación de las correspondencias desde las coordenadas x,y en una imagen con destino a los URL. Ahora se desaprueba porque no resulta eficaz y no funciona bien con los navegadores no GUI, pero puede que algunos sitios aún lo usen. Los mapas de tipo servidor en *IIS* adoptan la forma de archivos con la extensión “.map”, y su formato no es compatible con los archivos de *Apache* equivalentes.

El mejor planteamiento es convertir cualquier mapa de tipo servidor en un mapa de tipo cliente ya que así se proporciona una mayor experiencia en cuanto al navegador para el usuario. Si esto no es posible, se puede usar un simple *script* Perl para convertir los archivos en un formato que se pueda usar en *Apache*.

## 3. Conexiones entre *scripts* y bases de datos

Puede ser que los sitios más complejos tengan páginas dinámicas basadas en el acceso a las bases de datos y la información en código máquina. La mayoría de los sitios *IIS* usan ASP (Active Server Pages o Páginas Activas de Servidor) como marco para la información en código máquina, y podrían usar *Access* o *SQL Server* para la base de datos, dependiendo del tamaño de la aplicación.

Hay muchas formas de manejar la migración de los *scripts* ASP. Algunos de los más populares son los siguientes:

1. *Chili!Soft* Paquete ASP para Unix (ahora llamado *Sun ONE Active Server Pages*).
2. *ASP2PHP*
3. Módulo *Apache::ASP* de *Apache*
4. Conversión manual a un nuevo lenguaje.

*Chili!Soft* ASP es un producto propietario, pero en algunos casos podría proporcionar una ruta de migración muy eficaz en cuanto a costes.

*ASP2PHP* es un convertidor de *scripts* autónomo que convierte archivos de texto escritos en ASP y *VBScript* a archivos de texto escritos en PHP. Se está elaborando el pertinente soporte para los archivos ASP que usan *JScript*. PHP es un marco de *web-scripting* muy popular con cierto parecido a ASP, de manera que los creadores deberían considerarlo como una fácil transición. En el caso de proyectos de más envergadura, suele ser mejor tener una separación



mayor entre el diseño de la página y la lógica del *script* que la permitida por los modelos en el caso de los modelos PHP o *ASP*. En estos casos, la mejor opción podría ser una conversión manual usando un sistema de plantillas.

*Apache::ASP* proporciona dispositivos de tipo *ASP* directamente a través del marco *Apache*, junto con *scripting* en Perl. *VBScript* y *JScript* no tienen soporte.

En algunos casos puede ser mejor considerar una conversión manual desde *ASP* a un nuevo marco. Esto permite una flexibilidad máxima, y los sitios complejos pueden beneficiarse de su traslado a un sistema de plantillas como *Template Toolkit* (<http://www.tt2.org/>).

Todos los sistemas de *scripting Apache* tienen facilidades de acceso a las bases de datos para un amplio abanico de tipos de bases de datos (SQL, de archivo plano, indexada, LDAP, NIS, etc.) de modo que se pueden construir sitios dinámicos, de cualquier complejidad, manejados por datos.

#### 4. Ampliaciones de FrontPage

El paquete de diseño web *FrontPage* presentaba un conjunto de extensiones para permitir la gestión remota del contenido web. Desde entonces las han utilizado otros paquetes de diseño web.

Las extensiones de *FrontPage* están disponibles para los sistemas Unix, pero no siempre son apreciadas por los administradores de *Apache* por muchos motivos como, entre otros, los asuntos de seguridad y el gran número de cambios que introducen en el área estándar de almacenamiento de páginas web.

Ahora se dispone de una sustitución basada en los elementos estándar en forma de protocolo WebDAV (RFC2518). Tiene el soporte de la mayoría de los servidores (entre otros, *Apache*, usando el módulo *mod\_dav*) y ahora es el protocolo de gestión de sitios de web preferido. Microsoft ha dado apoyo a WebDAV en su serie de *Office* desde *Windows 2000* y también se puede acceder al mismo directamente a través de *Windows Explorer*, de modo que un servidor Linux/Unix/*Apache* puede dar soporte tanto a clientes propietarios como OSS usando el mismo mecanismo.

##### 14.7.1.2. Migración de un sitio web estático

En este ejemplo se ve el proceso completo para transferir un sencillo sitio web estático de *IIS* en *Windows NT* a *Apache* en GNU/Linux.

1. Prepare el servidor GNU/Linux, conéctelo a la red y pruebe *Apache*. La mayoría de las distribuciones GNU/Linux proporcionan paquetes de *Apache* preconfigurados, por lo que normalmente es sencillo. Un servidor visible en Internet va a necesitar, sin duda alguna, reforzar su seguridad antes de conectarse.
2. Localice los datos del sitio web en el servidor *IIS* (normalmente en **C:\InetPub**) y tenga una copia lista para ser transferida, por ejemplo usando un paquete de archivos Zip.
3. Copie el archivo Zip en la máquina GNU/Linux (por ejemplo, usando FTP) y desbloquéelo en la ubicación elegida para los datos del sitio web. Se configura como **DocumentRoot** en el archivo **httpd.conf** de *Apache*, y está normalmente en algún lugar como **/var/www/html**.

4. Edite **httpd.conf** y añada **default.htm** a la cláusula **DirectoryIndex**. (Se ha acordado configurar a *Apache* para buscar páginas iniciales/por defecto con el nombre de **index.html**, mientras que *IIS* utiliza **default.htm**, lo que permite usar los dos nombres.
5. En esta etapa, el sitio debería empezar a funcionar, aunque se debe acceder a él por el nombre del nuevo servidor más que por el oportuno URL. También puede presentar problemas donde los datos del sitio hacen un uso poco coherente de las mayúsculas y las minúsculas en los nombres de archivos y los URL, y donde se ha usado “\” en los URL.
6. De ser posible, pruebe el sitio en esta etapa y corrija cualquier problema editando los datos del sitio. Esto dará los mejores resultados. Hay herramientas para control automático que recorrerán el sitio y nos dirán si algún enlace indica ubicaciones no disponibles. También podría hacer una lista de páginas a las que no se llega en esta etapa, y ejecutar cada página a través de un controlador de HTML.
7. Si no es factible fijar los datos del sitio, añada estas líneas de configuración a **httpd.conf**:

```
LoadModule spelling_module modules/mod_spelling.so
AddModule mod_spelling.c
CheckSpelling on
```

Observe que esto provoca un barrido de directorio y una redirección de HTTP para cada parte con las mayúsculas mal puestas/faltas de ortografía de un URL, por lo que ha de vigilar las cuestiones de rendimiento.

8. Las páginas que usan incorrectamente “\” de los URL se pueden manejar con **mod\_rewrite**, añadiendo estas líneas a **httpd.conf**:

```
RewriteEngine on
RewriteRule ^(.*)\\(.*)$ $1/$2 [N]
```

Aquí se sustituye la primera \ por / en el URL y luego se repite por si hay más de una \.

9. Compruebe los mapas de imágenes del servidor usando un comando como el siguiente:

```
Find /var/www/html -name '*map*' -print'
```

Edite a mano si solo se trata de uno o dos, o use un *script* para fijarlos/corregirlos si son muchos.

10. En este punto, todo el sitio debería funcionar correctamente. Quizás quiera configurar FTP, *Samba* o WebDAV para proporcionar acceso a las páginas de actualización.
11. Para poner en producción el sitio, desconecte el antiguo servidor y cambie la dirección IP de la nueva máquina para sustituirla, o cambie la entrada DNS del sitio web para indicar el nuevo servidor.

#### 14.7.1.3. Una sencilla configuración WebDAV

WebDAV puede usarse para gestionar el contenido de alguno de sus sitios web o de todos ellos. En este ejemplo se usa para todo el sitio, por lo que no se debería permitir ningún otro acceso. (Otros sistemas de gestión, como el FTP o el acceso

directo los archivos confundirá a los clientes de WebDAV ya que no usan el mismo esquema de bloqueo.

1. Haga un directorio para los bloqueos de WebDAV. Debería pertenecer al mismo usuario y grupo que *Apache* ejecuta como tal (véase las opciones de configuración para **Usuario** y **Grupo** en **httpd.conf**). Una buena elección sería **/var//httpd/webdavlocks**.

2. Añada estas líneas a la parte principal de **httpd.conf**:

```
LoadModule dav_module lbexec/libdav.so
AddModule mod_dav.c
DAVLockDB /var/httpd/webdavlocks
```

3. Encuentre la sección **Directory** o **Location** que tiene que ver con el sitio web erróneo y añada líneas como ésta:

```
DAV On
AllowOverride None
Options Indexes
AuthType Basic
AuthName "Website Managers Only"
AuthUserFile /var/httpd/htpasswd
<LimitExcept GET HEAD OPTIONS>
    require valid-user
</LimitExcept>
```

4. Asegúrese de que los directorios y archivos relacionados pertenecen al mismo grupo y usuario en que se ejecuta *Apache*, usando un comando con esta forma:

```
Chown -R apache:apache /var/www/html
```

5. Cree el archivo de contraseñas:

```
touch /var/httpd/htpasswd
chown rootapache /var/httpd/htpasswd
chmod 640 /var/httpd/htpasswd
```

6. Cree una contraseña para un usuario llamado *webadmin* (o cualquier otro nombre que usted elija):

```
htpasswd -m /var/httpd/htpasswd webadmin
```

7. Reinicie *Apache* o haga que relea sus archivos de configuración, por ejemplo:

```
/etc/INIT.d/httpd reload
```

8. Ahora ya puede gestionar todo el sitio usando el protocolo WebDAV. *Windows 2000* y los clientes más recientes pueden acceder al mismo como "Lugar de Red" en *Windows Explorer*, y las aplicaciones de *Office* pueden guardar datos directamente en el sitio. GNU/Linux proporciona funciones similares a través de **davfs**.

9. Observe que el esquema aquí descrito sólo da una limitada seguridad. Debería leer el manual de *Apache* donde encontrará más información sobre la autenticación del usuario y así podrá elegir un esquema adecuado a sus necesidades. Puede ser necesario usar SSL para garantizar las transacciones; esto se puede hacer con **mod\_ssl** de *Apache*.

### 14.7.2. Bases de datos: migrar de Access y SQL Server a MySQL o PostgreSQL

Muchos proyectos pequeños de bases de datos en *Windows* usan *Access*. Este resulta un producto atractivo para mucha gente pues es bastante sencillo para comenzar, y tiene una interfaz de usuario familiar. Pero *Access* tiene fuertes limitaciones, al no haber sido diseñado para un entorno multiusuario pesado y no puede hacer frente a grandes conjuntos de datos.

Las bases de datos más grandes podrían usar *SQL Server* o una de las bien conocidas bases de datos relacionales: *Oracle*, *Sybase*, *DB2*, etc. En el caso de estos grandes sistemas, puede que lo mejor sea dejar que la base de datos funcionen en la plataforma existente y sólo transferir las aplicaciones de cliente a plataformas OSS. Esto va especialmente bien cuando la Administración tiene habilidades profundas para la base de datos existente y está usando muchos dispositivos propietarios. Hay modos estándar de conectarse a bases de datos relacionales en red, así que la elección de plataforma puede ser diferente para la base de datos y las aplicaciones de cliente. Además, la mayoría de las bases de datos propietarias que no son de Microsoft están disponibles en plataformas GNU/Linux y Unix, por lo que es posible cambiar el sistema operativo sin tener que aprenderse una base de datos totalmente nueva.

Por otra parte, las bases de datos propietarias pueden resultar muy caras por lo que vale la pena considerar si un producto OSS podría hacer el trabajo con eficacia.

Las dos bases de datos más conocidas son *MySQL* y *PostgreSQL*. Las dos son productos maduros y cuentan con una buena base de instalaciones y equipos creativos muy activos. Las dos tienen buen soporte para el SQL estándar y pueden dar un magnífico rendimiento.

También vale la pena recordar que las bases de datos no tienen que ser relacionales. Algunas tareas van mejor con otros modelos, y el uso directo de un producto OSS como la *Berkeley DB* de *Sleepycat* puede ser extremadamente eficaz. Igualmente, el modelo LDAP de bases de datos jerárquicas en red es muy adecuado para algunos tipos de aplicaciones distribuidas.

#### 14.7.2.1. Migración de las bases de datos de Access

*Access* sólo está disponible en plataformas *Windows*. Por ello, esas bases de datos deben ser transferidas a algún otro paquete si se proyecta un entorno completamente OSS. Un interesante y útil escenario intermedio implica la migración de los datos a una base de datos OSS, pero sin dejar de usar *Access* como entorno frontal. Esto tiene la deseable propiedad de eliminar muchas de las restricciones y problemas del almacén de datos *Access*.

##### 1. Importación/exportación manual

Hay varios modos de transferir los datos desde *Access* a otras bases de datos. En el caso de conjuntos de datos sencillos, quizás lo más fácil sea exportar las tablas desde *Access* como archivos CSV (Valores Separados por Comas) y luego importar éstas al nuevo servidor. Este método requiere que las tablas se creen manualmente primero en el nuevo servidor, pero no necesita ningún software especial.

Como ejemplo, estos son los comandos para crear una base de datos con una sencilla tabla e importar un archivo CSV a *MySQL*. En primer lugar introduzca lo siguiente en un punto indicativo de shell:

```
Mysql --user=myusername -p
```

Luego introduzca lo siguiente:

```
create database mydb;
use nydb;
create table mytable (
    firstname      char (30),
    surname        char (30),
    postcode       char (19)
);
load data local infile 'exportfile.csv'
into table mytable
fields terminated by ',' enclosed by '*'
lines terminated by '\r\n';
```

## 2. Importación/exportación “scripted”

Existen varios programas y *scripts* que exportarán una base de datos *Access* que tienen toda la información necesaria para recrear las tablas en otro gestor de bases de datos. Algunos producen archivos para ser copiados en una nueva plataforma, mientras otros se conectan directamente en la red y hacen los cambios de inmediato. Un ejemplo de los *scripts* para escritor de archivos es **exportsql2.txt**, disponible en <http://www.cynergi.net/exportsql>, que produce archivos con cláusulas DROP TABLE, CREATE TABLE e INSERT que reproducirán la base de datos *Access* en *MySQL*.

En el informe de Paul DuBois titulado *Migrating from Microsoft Access to MySQL* se describen otras herramientas de migración (véase la página web <http://www.kitebird.com/articles/access-migrate.html/>).

Una vez que los datos han sido transferidos ya se puede seguir usando *Access* como entorno (frontal) borrando las tablas localmente y enlazando con las tablas recién creadas en el servidor *MySQL*.

### 14.7.2.2. Migración de las bases de datos de SQL Server

El proceso aquí es en gran medida igual al descrito anteriormente; en el caso de las bases de datos sencillas suele bastar con exportar los datos a un formato común (normalmente CSV) y luego importarlos a la nueva base de datos. Las bases de datos más complejas que incluyen procedimientos almacenados y disparadores necesitarán más esfuerzo, y en estos casos vale la pena mirar la gama de herramientas disponibles para ayudar al proceso de migración. Algunas son OSS y otras comerciales. He aquí algunos ejemplos:

1. *PGAdmin* es software libre para administrar bases de datos *PostgreSQL*. Hay utilidades modulares para ellas que manejan la migración de datos de otros gestores de bases de datos. Más información en <http://www.pgadmin.org/>.
2. *SQLPorter* de Realsoftstudio: un producto comercial disponible en diversas variantes según los gestores de bases de datos inicial y final. En <http://www.realsoftstudio.com/overview.php> encontrará más información.
3. *SQLways* de Ispirer: un producto comercial que da soporte a una amplia gama de gestores de bases de datos. Más información en <http://www.ispirer.com/productos>.

4. *SQLyog* es otra herramienta comercial que gestiona en *MySQL* y también maneja la migración de datos desde otras bases de datos de tipo ODBC. Véase <http://www.webyog.com/sqlyog>.
5. El sitio web de *MySQL* da una lista amplia de otras herramientas de conversión (<http://mysql.com/portal/software/convertors/index.html>).

#### 14.7.2.3. Temas de migración de bases de datos

A veces transferir los datos es la parte más fácil del trabajo, aunque si el acceso a los mismos es por la red como tablas SQL entonces hay mucho más que hacer.

Lo más probable es que los problemas vengan de las utilidades secundarias y lenguajes *scripting* que rodean a cualquier base de datos práctica. El propio SQL está estandarizado, aunque casi todos los vendedores de bases de datos lo amplían y animan a la gente a usar sus ampliaciones no estándar. También suele haber otros modos de conseguir un resultado dado en SQL, y elegir cuál es el más eficaz varía de una base de datos a otra.

Muchas aplicaciones para bases de datos se construyen con generadores de aplicaciones o constructores de formularios Y puede que no funcionen con otra base de datos que no sea aquella con la que venían.

Tanto *MySQL* como *PostgreSQL* han evolucionado muchísimo en los últimos años, por lo que es importante que se asegure de leer revisiones recientes al estudiar cuál usar y si hacer la migración o no.

#### 14.7.3. Programas de grupo: salirse de Exchange

*Exchange* proporciona servicios de libreta de direcciones, calendario y correo electrónico. Normalmente se usa con el cliente *Outlook* en *Windows*, aunque algunas instalaciones también usan *Outlook Web Access (OWA)* para proporcionar funciones básicas a través de la interfaz de web.

Todas las funciones de *Exchange* se pueden sustituir por paquetes OSS, a menudo con excelentes resultados. Los problemas surgen al tratar de proporcionarlas sin fisuras a clientes *Outlook*, ya que el mecanismo de comunicación entre *Exchange* y *Outlook* es propietario. *Outlook* es capaz de acceder a ciertos servicios abiertos basados en estándares, aunque en algunos casos la experiencia del usuario es diferente a la encontrada al usar el protocolo propietario. Así pues, vale la pena decidir desde el principio si hacer la transferencia a un paquete de cliente OSS al mismo tiempo que se hace la del servidor, dado que la población de usuarios verá algunos cambios incluso si se quedan con *Outlook*. El cliente más obvio para la sustitución es *Evolution* de Ximian.

##### 14.7.3.1. Temas generales

Todos los usuarios de *Exchange* tendrán nombres de usuario y contraseñas almacenados en el sistema. Las versiones recientes de *Exchange* usan Active Directory para ello, de modo que las notas que aparecen en cualquier lugar de este informe sobre los datos de registro de los usuarios en transferencia también se aplican a *Exchange*. En resumen, los servidores de tipo OSS pueden acceder a los datos de registro a través del LDAP, de manera que los nuevos usuarios pueden usar el Active Directory existente o se pueden transferir los datos a un almacén de datos de tipo OSS como *OpenLDAP*.

#### **14.7.3.2. Correo**

Los usuarios pueden almacenar grandes cantidades de correo, tanto personal como compartido con otros miembros del grupo. Puede haber requisitos de procedimiento o legales para mantener un diario de todo el correo recibido y enviado, en cuyo caso el almacenamiento y el acceso a estos datos debe tenerse en cuenta. Quienes tienen ordenadores portátiles pueden descargar todo su correo al laptop, o bien mantener una copia sincronizada en el máster que hay en el almacén central.

Al planear una migración a servicios de tipo OSS, es importante ubicar todos los datos guardados y asegurarse de que se podrá acceder a ellos una vez hecha la transición.

*Exchange* puede usar grupos *Windows* como listas de distribución: son los mismos grupos que utiliza el propio *Windows* para el control de acceso. Este no es el modo corriente de mantener las listas de distribución en un entorno OSS, pero se puede hacer así si se desea.

Si se está reteniendo a *Outlook* como cliente de correo, habrá que reconfigurarlo para usar IMAP en lugar de buzones de acceso "nativos".

*Exchange* no tiene facilidad para exportar así que la migración de datos se debe hacer a través de una conexión a cliente.

En las Secciones 11.2 y el Anexo C se da más información sobre los sistemas de correo OSS.

#### **14.7.3.3. Carpeta de direcciones**

Los usuarios de *Outlook* construyen automáticamente una carpeta de direcciones ya que envían y reciben mensajes. También tienen acceso a una o más carpetas de direcciones si usan el servidor *Exchange*. Los contenidos de estas libretas de direcciones deben transferirse a una forma que pueda leerse con OSS. Las carpetas de direcciones personales pueden exportarse en forma vCard, lo que muchos pequeños clientes de correo ven bien y puede ser analizado por *scripts* para su conversión a otros formatos si fuera necesario. Igualmente, las carpetas de direcciones compartidas se pueden exportar y luego cargar en un almacén de LDAP.

Los principales problemas pueden venir del hecho de que *Outlook* y *Exchange* tienden a no usar direcciones de correo RFC822 estándar internamente, de manera que es posible que los datos de las carpetas de direcciones no incluyan direcciones utilizables cuando se exportan. En este caso será necesario algún proceso posterior mediante un *script* con acceso al almacén del Active Directory para traducir las direcciones que están "en forma interna" a direcciones RFC822 estándar. Esta traducción podría ser necesaria incluso en el caso de que *Outlook* se retuviera como pequeño cliente, ya que no podrá usar direcciones "en forma interna" al enviar correo a protocolos basados en estándares como SMTP.

#### **14.7.3.4. Calendario**

Algunas Administraciones usan mucho las facilidades de calendario de *Outlook* para disponer reuniones y gestionar reservas de alojamiento. Estas facilidades se pueden usar sin *Exchange*, pero con algunas limitaciones.

Si se ha planeado una migración simultánea a clientes OSS, se deberían exportar los calendarios en forma vCal y pasarlos a la nueva plataforma de gestión de calendarios.

## **14.8. Migración de aplicaciones del escritorio a OSS**

### **14.8.1. Ofimática**

#### **14.8.1.1. Conversión de documentos**

*OpenOffice.org* puede leer y escribir notablemente bien formatos de Microsoft, por lo que no es necesario convertir los documentos durante el proceso de migración. Si se desea hacer una conversión de documentos, se puede automatizar con el dispositivo Autopilot seleccionado del menú de Archivo de *OpenOffice.org*. Éste proporciona un modo de convertir documentos en masa. La decisión de convertir depende del uso futuro del documento. En el Capítulo 5 se habla en términos generales de los formatos de documentos y la conversión. Si los documentos se van a editar repetidamente, el formato debería ser el usado por la mayoría de los editores.

#### **14.8.1.2. Conversión de plantillas**

*OpenOffice.org* puede usar plantillas directamente en formato *Word 97*, pero en la práctica es mejor no convertirlas a plantillas nativas y guardarlas en la oportuna área de plantillas compartida, que brinda la oportunidad de testar cada plantilla y corregir cualquier error de conversión. *OpenOffice.org*, por su parte, hace la mayoría del trabajo de conversión, y el proceso se debe automatizar para grandes grupos de plantillas que usan la función de Conversión de Documentos Autopilot que viene en el menú Archivo.

#### **14.8.1.3. Conversión de macros**

*OpenOffice.org* usa un lenguaje para macros tipo BASIC. Estructuralmente es muy similar a los lenguajes usados por *Word* y las últimas versiones de *WordPerfect*. Pero los nombres de los objetos con los que trabaja son diferentes, con lo que las macros necesitarán un cierto esfuerzo de conversión manual.

Las macros en los documentos son un grave riesgo de seguridad y no son necesarios en la mayoría de las tareas cotidianas, por lo que vale la pena ver si se pueden obviar. La mayoría de los riesgos de formateo se manejan mejor usando plantillas y estilos, y la manipulación de datos se puede hacer mediante formularios.

Las versiones de *OpenOffice.org* desde la 1.1. incluyen un grabador de macros, que facilita la creación de macros simples si resultan ser fundamentales.

Por el momento no hay ningún modo automatizado de convertir las macros, aunque se está trabajando en ello.

#### **14.8.1.4. Procesadores de textos**

Hay muchos paquetes de proceso de textos en uso en los sistemas *Windows*. Es muy probable que las entidades bien gestionadas hayan estandarizado un paquete, o puede que estén pasando de uno a otro. Los paquetes más corrientes son:

- *Microsoft Word*
- *Microsoft Works*



- *WordPerfect*
- *Lotus AmiPro* y *Lotus WordPro*
- *IBM DisplayWrite*

La opción de migración a OSS recomendada es *OpenOffice.org*.

Los archivos en *Microsoft Works*, *IBM DisplayWrite* y los formatos Lotus no los puede leer directamente *OpenOffice.org*, por lo que hay que convertirlos. A menudo es posible exportar los archivos desde la respectiva aplicación a algún formato común aceptable; la conversión también puede requerir la herramienta de un tercero.

Los archivos de *WordPerfect* no se pueden leer directamente, pero hay un proyecto en curso para incluir este formato en *OpenOffice.org*. Existe un programa de conversión basado en *scripts* que podría usarse para la conversión de formatos masivos.

#### **14.8.1.5. Autoedición**

La producción de documentos más allá de la capacidad de los procesadores de texto se suele hacer con paquetes de autoedición (*Desktop Publishing -DTP*). Los más corrientes son:

- *Framemaker*
- *Pagemaker*
- *QuarkXPress*

El producto OSS *Scribus* de <http://web2.altmuehlnet.de/fschmid> persigue reemplazar a estos paquetes y puede valer la pena tenerlo en cuenta.

*OpenOffice.org* tiene mucha más capacidad que los procesadores de texto en la época en la que aparecieron los primeros paquetes de autoedición. Características avanzadas como los documentos maestro han posibilitado el manejo de grandes proyectos como la producción de libros, y las características de configuración de páginas son ahora más flexibles.

Otros planteamientos alternativos contemplan el uso de paquetes de post-proceso en los que el texto se marca en un lenguaje similar al HTML y luego se convierte a su configuración final para impresión aplicándole hojas de estilo. Los sistemas sin GUI pueden resultar muy útiles para producir documentos que cambian rápidamente y para la impresión a petición desde material procedente de bases de datos.

#### **14.8.1.6. Hojas de cálculo**

Entre las hojas de cálculo corrientes de *Windows* tenemos las siguientes:

- *Microsoft Excel*
- *Lotus 123* y derivadas

*Excel* es la más usada con mucho.

La opción OSS recomendada es *OpenOffice.org* aunque *Gnumeric* también podría tenerse en cuenta.

En la mayoría de los casos, una migración desde *Excel* o *Lotus 123* a *OpenOffice.org* o a *Gnumeric* plantearía pocos problemas a no ser que las hojas de cálculo contengan controles u otros mecanismos que requieran macros. En este caso, se deben reescribir tales controles y macros.

#### 14.8.1.7. Gráficos para presentaciones

En un entorno *Windows*, las presentaciones se suelen crear mediante *Microsoft PowerPoint* o *Corel Draw*. *PowerPoint*, con su formato de archivo **\*.ppt** es el más corriente.

La opción OSS recomendada es *OpenOffice.org*. Puede leer *slideshows* y plantillas con muy pocos errores, y se puede configurar para escribir archivos **\*.ppt** si se desea. Como ya se ha dicho en la Sección 14.8.1.2, valdría la pena trasladar en masa las plantillas importantes al formato nativo *OpenOffice.org*.

Los usuarios deberían poder pasar de *PowerPoint* a *OpenOffice.org* y viceversa fácilmente ya que los conceptos y la configuración de pantalla son muy similares.

#### 14.8.1.8. Manipulación de imágenes y gráficos

Los paquetes gráficos se dividen en tres categorías principales:

- Gráficos para presentaciones, de los que se trató en la Sección 14.8.1.7.
- Gráficos vectoriales, tipificados por los programas y paquetes MCAD de gama baja como *Microsoft Visio*.
- Gráficos de mapa de bits, incluidos los programas de dibujo (*paintbrush*) y los paquetes para el tratamiento de fotografías como *Adobe Photoshop*.

### 1. Aplicaciones de gráficos vectoriales

*Open Office.org* incluye un dispositivo para dibujo.

*Dia* (<http://www.lysator.liu.se/~alla/dia>) es un paquete OSS similar a *Visio*. Se usa mucho para generar diagramas en documentación, y tiene filtros que leerán los archivos de las antiguas versiones de *Visio* (no de la versión 2002). Existen librerías de símbolos para algunas aplicaciones. *Kivio* hace un trabajo similar y está pensado para integrarse bien en un entorno de KDE, pero parece más centrado en los diagramas de flujo. *Sodipodi* (<http://sodipodi.sourceforge.net>) trabaja bien con SVG (Gráficos redimensionables para vectores).

Los archivos creados en *Visio* y paquetes similares se pueden leer con software OSS, pero se debería probar en cada caso individual antes de planificar una migración.

### 2. Gráficos de mapa de bits

Esta categoría va desde sencillos programas como *Paint* hasta los que hacen tratamiento de imagen avanzado como *Adobe Photoshop*. El mundo del OSS ha producido por lo menos tantos programas gráficos como el sector del software propietario, y con características y calidad igualmente variables. Aún así, hay un paquete, *The Gimp*, que destaca sobre los demás (<http://www.gimp.org/>).

*The Gimp* puede leer casi todos los formatos de archivos para gráficos de mapa de bits conocidos (incluido el formato interno propio de *Photoshop*) y también

puede generar la mayoría de ellos. Proporciona todas las características de un buen programa de dibujo junto con capas, canales, y otras herramientas avanzadas bien conocidas por los usuarios de *Photoshop*. *The Gimp* se usa mucho para generar y mejorar imágenes para la web y para ser publicadas. La única característica que falta actualmente es la gestión del color, por lo que no resulta apto para trabajos de alta calidad.

#### 14.8.1.9. Generar PDF

Generar un archivo PDF en OSS es mucho más fácil que con *Windows* donde hay que adquirir un software como *Adobe Acrobat*. Para esto hay además diversas herramientas de PDF y Postscript disponibles en distribuciones estándar. Adicionalmente, *OpenOffice.org* ofrece la posibilidad de generar directamente salida en formato PDF. La producción de un PDF puede configurarse como un servicio para impresión, y así facilita la continuidad de los usuarios de *Windows*.

#### 14.8.2. Correo

Hay una enorme variedad de interfaces de usuario para correo electrónico, tanto para entornos OSS como propietarios. Así pues, este documento sólo puede dar una visión general esquemática de los temas y el proceso de migración. En la Sección 11.2 y en el Anexo C también se trata el tema del correo.

Los principales temas relacionados con los clientes son éstos:

- La elección de un nuevo Agente de Usuario de Correo y en consecuencia de la interfaz de usuario.
- La migración del correo personal almacenado que exista.
- La migración de las entradas existentes en la libreta (carpeta) de direcciones.

Cualquiera que sea el MUA elegido, habrá que transferir el correo guardado y las entradas de la carpeta de direcciones- Si el antiguo MUA estaba configurado para guardar todas las carpetas en un servidor IMAP, el trabajo por hacer será poco y sólo hay que configurar el nuevo MUA para acceder a las mismas in situ. Cuando los archivos locales se han usado como carpetas, será necesario rastrearlos y convertirlos. Por defecto, *Outlook* guarda el correo en archivos con la extensión **\*.pst\*** en

**C:\Documents and Settings<username>Local Settings\Application Data\Microsoft\Outlook.**

Estas son algunas herramientas útiles para la migración:

- <http://outport.sourceforge.net/> - exporta los datos de *Outlook* a *Evolution*, etc. Está en pleno desarrollo y a mediados del 2003 aún tiene importantes limitaciones como, por ejemplo, la incapacidad de exportar archivos adjuntos de correo.
- Herramienta de exportación propia de *Outlook*, posiblemente CSV de escritura o **formato Excel**. También adolece de no poder exportar archivos adjuntos a estos formatos.
- <http://sourceforge.net/projects/ol2mbox> - herramienta OSS para convertir los archivos **\*.pst\*** de *Outlook* en formatos utilizables por los usuarios de correo OSS. Soporta archivos adjuntos.
- *Kmailcvt* - herramienta OSS para convertir algunos formatos propietarios para su uso con *Kmail*

### 14.8.3. Calendarios y Programas de grupo

Los calendarios, junto con la gestión de contactos y el correo, suelen agruparse con el encabezamiento general de Gestión de la Información Personal (PIM). Algunos paquetes integrados, como el *Outlook* de Microsoft, proporcionan las tres funciones en una sola interfaz, pero otros como *ACT!* se concentran en la gestión de contactos y podría verse como más cercano a los sistemas CRM (Gestión de las Relaciones con el Cliente).

La opción final recomendada OSS es *Evolution*, que integra las funciones en gran medida igual que lo hace *Outlook*. *Mozilla* también podría tenerse en cuenta; incluye un cliente de correo electrónico con capacidad y ahora hay un módulo de calendario que se puede encontrar en <http://www.mozilla.org/projects/calendar> - se basa en el estándar abierto iCalendar y los usuarios pueden publicar y compartir calendarios usando el protocolo WebDAV.

#### 14.8.3.1. Calendarios

Algunas de las mejores utilidades para crear calendarios en OSS están en los paquetes de *groupware* en web y vale la pena considerarlas como posibles servicios a nivel de empresa.

Las normas del iCalendar (anteriormente vCalendar) definen un formato de intercambio para las entradas de calendario. En <http://www.imc.org/pdi> y en RFC2445, RFC2446 y RFC2447 se puede encontrar más información al respecto. La mayoría de las series de calendarios OSS pueden manejar datos en este formato, por lo que es la ruta de migración preferida así como el medio normal para la gestión cotidiana de calendarios.

#### 14.8.3.2. Gestión de contactos

Casi todos los paquetes pequeños que han existido han definido su propio formato para el almacenamiento de los datos en la libreta de direcciones. Muchos están limitados a guardar sólo direcciones de correo electrónico, pero los formatos más recientes tienden a incluir todo tipo de información sobre contactos. Esta diversidad de formatos hace que la migración sea más difícil de lo que lo sería de otro modo.

Afortunadamente, en los últimos años las principales aplicaciones de correo electrónico en el mundo del OSS y el propietario han tendido a poner en marcha los formatos de intercambio en iCard (anteriormente vCard). Lo específico de este formato es que es abierto, y se puede obtener en <http://www.imc.org/pdi> y también en RFC2445 y RFC2446. Si se tienen que transferir los pormenores de los contactos desde una aplicación propietaria a una aplicación OSS, éste es el formato preferido.

Otro modo de manejar la información sobre contactos es consolidarla en un directorio de tipo empresa y luego acceder a él a través del LDAP. Ciertamente, esto se haría con datos de amplio uso como los del listín telefónico interno y la lista de correo electrónico que muchas empresas llevan. No constituye la sustitución total de las libretas de direcciones personales: una libreta de direcciones debería ser pequeña y centrada en las necesidades de la persona que la usa, mientras que un directorio será amplio y (probablemente) demasiado grande para navegar por él eficazmente.

Algunas de las herramientas para la migración mencionadas en la Sección 14.8.2 también pueden extraer información de los archivos de datos de *Outlook* en formato iCard.

#### 14.8.4. Navegación en la web

Lo más probable es que los usuarios de *Windows* usen alguna versión de *Microsoft Internet Explorer* para navegar en la red. También es posible que algunos usen *Netscape*, *Mozilla* u *Opera*. La opción de migración a OSS recomendada es *Galeon*, aunque *Mozilla* también podría considerarse porque funcionará con *Windows*.

La migración desde un navegador de red a otro resulta bastante fácil para los usuarios, ya que todos tienen características e interfaces de usuario similares (aparte de navegadores en modo texto como *Linux*, por razones obvias). Los temas que importan a los usuarios particulares se centrarán alrededor de la conversión de marcadores: la mayoría de los navegadores OSS pueden importar marcadores de IE y Netscape si están instalados en la misma plataforma, pero si el Sistema Operativo también se está transfiriendo, entonces puede ser necesario exportar primero los marcadores en forma de archivo HTML.

Toda empresa u organización que use páginas web de intranet debería comprobar si el HTML se ajusta a las normas W3C de modo que se visualice correctamente en todos los navegadores. Hay herramientas en <http://www.w3c.org/> que pueden ser de ayuda en este tema.

Toda página que dependa de JavaScript necesitará someterse a una prueba especialmente cuidadosa, ya que el dialecto varía de un navegador a otro y el uso de extensiones no estándar causará problemas.

Toda página que dependa de controles ActiveX necesitará ser rediseñada para trabajar de otro modo, ya que los navegadores OSS no admiten esta tecnología propietaria. ActiveX tiene un modelo de seguridad muy débil, por lo que inutilizarlo es, en cualquier caso, una buena medida.

Los navegadores OSS admiten bien los formatos accesorios de web más corrientes, como Java, PDF, Flash y Real Player (aunque a menudo usando un código no OSS para ello que a veces resulta difícil de encontrar en la página web del proveedor). Otros formatos como Shockwave Director habrán de usar el *CrossOver Plugin* de CodeWeavers.

#### 14.8.5. Bases de datos personales

Quienes manejan conjuntos de datos demasiado amplios o complejos para una hoja de cálculo pero no lo suficiente como para justificar una base de datos comercial completa suelen usar *Microsoft Access*. Este paquete proporciona un sencillo almacén de datos relacional junto con herramientas de construcción de formularios y *scripting*.

La migración de datos a las bases de datos OSS se explica en la Sección 14.7.2.

Hay ventajas en guardar las bases de datos sobre servidores bien gestionados incluso si la función central de TI no gestiona los datos ni admite las aplicaciones. Una migración OSS brinda la oportunidad de ofrecer ese servicio de almacenamiento de datos configurando un servidor en el que los particulares puedan construir sus propias aplicaciones de bases de datos. Hay varios paquetes basados en web que podrían elegirse para esto, como *PHPmyAdmin*; más información en <http://www.phpmyadmin.net/documentation>.

Las herramientas con GUIs más convencionales son:

- *Kexi* (<http://www.koffice.org/kexi>) - un entorno para bases de datos del proyecto KDE, dirigido a un mercado similar al de *Access*.
- *DBDesigner* (<http://www.fabforce.net/dbdesigner4>) - una herramienta para usuarios más avanzados que se integra tanto con Gnome como con KDE.
- *Knoda* (<http://www.knoda.org>) - otro sencillo entorno para KDE.

Ninguna de estas herramientas intenta leer los archivos de *Access*.

#### **14.9. Migración de los servicios de impresión a OSS**

En entornos pequeños de oficina es corriente que las impresoras se conecten directamente a las estaciones de trabajo en el escritorio. En las oficinas más grandes y las que necesitan un gran volumen de impresión, es más probable que usen impresoras en red, que pueden estar conectadas directamente a la red o controladas por un servidor de impresión.

Los entornos OSS admiten ambos modelos aunque es más corriente encontrar servidores de impresión y un pequeño número de impresoras de gran capacidad.

##### **14.9.1. El modelo de impresión de Windows**

Imprimir en *Windows* se hace casi siempre a partir de un menú dentro de una aplicación GUI. Es muy raro usar la línea de comandos. Las aplicaciones generan salidas impresas usando un proceso muy similar al que usan para poner información en la pantalla. Un conjunto de programas controladores concretos para la impresora son los usados por el sistema operativo para generar la secuencia de datos real para la impresora. Estos programas controladores los suele suministrar el fabricante de la impresora, y se deben instalar localmente o en el servidor de impresión antes de poder imprimir algo. En un entorno de red, lo mejor suele ser instalar y configurar los controladores en el servidor de impresión para que los clientes no necesiten ser configurados manualmente (sigue siendo necesario conectar las impresoras al cliente: esto se puede hacer a mano o mediante un *script* de conexión (archivo de comandos de conexión)).

##### **14.9.2. Unix y el modelo de impresión de GNU/Linux**

GNU/Linux heredó su modelo de impresión de BSD Unix. Las aplicaciones generan archivos o secuencias de datos de impresión que se pasan a una cola de impresión que se hace cargo de la tarea de impresión. Las tareas pueden tener que hacer cola y se pasan abiertamente a otras máquinas en la red. Los primeros sistemas Unix no tenían interfaz independiente de la impresora para generar datos de impresión, por lo que cada aplicación debía incluir un código para cada tipo de impresora que quisiera hacer funcionar. En los tiempos en que la impresión era sólo de caracteres, esto no era problema, pero cuando los fabricantes comenzaron a añadir facilidades para gráficos, cada uno de ellos creó un nuevo y diferente lenguaje de impresora.

Los sistemas de impresión BSD han tenido siempre la capacidad de suministrar tareas de impresión a través de un conjunto de filtros, con lo que la gente empezó escribir filtros que se convertían de un lenguaje de impresora a otro para ampliar la gama de impresoras admitidas. Muchas de las mejores impresoras utilizadas en laboratorios de investigación tienen intérpretes PostScript, así es que PostScript llegó a usarse como el lenguaje independiente de impresora más común.

Ahora la mayoría de los distribuidores de GNU/Linux están cambiando el sistema de impresión BSD por un nuevo paquete llamado *CUPS* (Sistema Común de Impresión



Unix), que admite el Protocolo de Impresión para Internet (IPP) además del protocolo tradicional **Ipr**. Y así se completa la transición al nuevo modelo de impresión:

- Las aplicaciones generan tareas de impresión en PostScript.
- Cuando las tareas se pasan al sistema de impresión, la aplicación puede requerir alguna característica especial que la impresora admita (doble impresión, plegado, cosido, perforación, encuadernación, etc.). Las demandas tienen un formato estándar, pero es obvio que sólo saldrán bien si la impresora dispone del hardware necesario. Hay una forma estándar para que las aplicaciones averigüen qué características admite una impresora dada.
- Las tareas pueden ser puestas en espera localmente en la estación de trabajo, o se las puede pasar inmediatamente a un servidor de impresión. El usuario no necesita conocer qué método se usa.
- El sistema de impresión puede distribuir tareas por varias impresoras similares en una "red" de impresoras.
- El servidor de impresión realiza la tarea a través de un conducto de filtros para convertirla en plataformas para cualquier formato que necesite la impresora, y para controlar la comunicación con ésta.

Hay en estos momentos más de 600 modelos de impresoras conocidos que trabajan perfectamente con este modelo (es decir, las aplicaciones GNU/Linux pueden acceder a todas las funciones disponibles usando los controladores *Windows* facilitados por el fabricante, y con resultados equivalentes o mejores).

Aunque PostScript es el formato intermedio más usado, *CUPS* se puede configurar para admitir casi cualquier formato de archivo para los que haya filtros. En particular, es corriente permitirle a PDF, JPEG y algunos otros formatos que se impriman directamente, y a algunos sitios y filtros que hagan impresiones automáticas del correo electrónico, etc.

*CUPS* proporciona interfaces compatibles con la serie **Ipr** de BSD y también con el **Ip** del Sistema V. Por lo tanto es posible sustituir los sistemas antiguos en su totalidad en las máquinas existentes (*FreeBSD*, *OpenBSD*) y las variantes de Unix más comerciales. Se está preparando un puerto para *Windows*.

*CUPS* proporciona una amplia gama de características y facilidades, incluido el auto-descubrimiento de servidores de impresión, contabilidad de páginas, cuotas, etc. Más adelante damos, para más información, los sitios web de *CUPS*.

### 14.9.3. Configurar un servicio de impresión basado en OSS

En los casos de despliegue muy pequeños, es sencillo configurar las impresoras unidas directamente en la estación de trabajo de cada cliente. Éstas se pueden compartir en toda la red, si se desea, y *CUPS* lo admite con facilidad.

El uso de servidores de impresión se recomienda en todos los casos en que hay más que unos pocos clientes, o cuando hay un importante volumen de impresiones. Se deberían instalar una o más máquinas de servidor de impresión, y dar nombres lógicos en el DNS además de sus nombres de anfitrión. Esto permite configuraciones a nombres de referencia como **prntserver.example.org** mejor que **pc35.example.org**, lo que facilita mucho el reorganizar el servicio posteriormente. Todas las máquinas de clientes deberían ser configuradas para referirse a uno de los servidores de impresión de cara a

todas las necesidades en este campo: esto evita tener que reconfigurar a los clientes cuando se añaden o quitan impresoras.

#### **14.9.4. Imprimir desde clientes de Windows a impresoras adscritas a GNU/Linux**

Hay varios modos de establecer servidores de impresión de tipo GNU/Linux que admitan máquinas de escritorios *Windows*. El esfuerzo inicial y el necesario por cliente es diferente en cada uno de ellos.

##### **14.9.4.1. Usando el protocolo lpr**

Este método es adecuado cuando un número muy reducido de clientes de *Windows* ha de ser admitido.

**lpr** es un protocolo muy corriente para pasar las tareas de una máquina a otra y viceversa. Como ya se ha dicho, está siendo sustituido poco a poco por el IPP, pero está muy implantado y se puede usar desde la mayoría de las versiones de *Windows*.

1. Asegúrese de que la máquina GNU/Linux está configurada para aceptar tareas usando **lpr**.
2. Consiga un buen conjunto de controladores (drivers) de *Windows* para la impresora. Lo mejor sería el controlador *CUPS* genérico que genera el PostScript portátil, pero es imposible usar controladores concretos para impresoras si *CUPS* está configurado para permitir impresión en bruto.
3. Conéctese a la máquina *Windows* como Administrador.
4. Abra la utilidad Networking en el Panel de Control, selecciones la tab de Servicios y asegúrese de que figura "Microsoft TCP/IP printing". Si es necesario añádala (será preciso el CD de distribución y un reinicio).
5. Configure la impresora en el cliente de *Windows* como si fuera una impresora local (ino en red!). Al seleccionar un puerto para la impresora cree un nuevo "puerto LPR" y configúrelo para enviar las tareas al servidor GNU/Linux.

El cliente *Windows* ya puede enviar tareas de impresión a la máquina GNU/Linux, pero puede que las herramientas de *Windows* no puedan ver y hacer las tareas en espera. *CUPS* admite la gestión basada en la red, de manera que se debería aconsejar a los usuarios que usen lo que necesitan.

##### **14.9.4.2. Usando impresoras compartidas**

Este método vale igualmente para un número reducido de clientes de *Windows*. Funciona con *Windows 95/98/ME* así como *Windows NT/2000*.

1. Instale y configure *Samba* en el servidor GNU/Linux. Siga las instrucciones para crear elementos compartidos de impresora: es fácil que *Samba* cree automáticamente una parte para cada impresora de la que el servidor tenga conocimiento.
2. En cada cliente de *Windows*, use el *Add Printer Wizard* para añadir una impresora para red. Debería poder navegar por una lista de servidores para encontrar una que quiera usar. Habrá de instalar localmente controladores de impresora en la máquina del cliente.



Como perfeccionamiento de este esquema, un administrador puede usar el ADD Printer Wizard para descargar controladores de impresora *Windows* al servidor *Samba*, de modo que no necesiten ser instalados de manera individual en los clientes.

#### 14.9.4.3. Usando configuración Point and Print (Señale e Imprima)

Este método es adecuado para grandes instalaciones y aquellas en las que un personal menos cualificado debe configurar nuevas máquinas clientes. Requiere más esfuerzo en principio para su instalación, pero es mucho más fácil de usar una vez instalada. El proceso es bastante complicado, por lo que resulta útil consultar los HOWTO de *Samba* donde encontrará más información al respecto.

1. Instale y configure *Samba* en el servidor GNU/Linux. El soporte pleno señale e imprima (*Point and Print*) lo da la versión 3.0 u otra superior, aunque la 2.2.4. admite muchas funciones.
2. Configure *CUPS* para soportar las impresoras *Windows* añadiendo el paquete de controladores *CUPS*.
3. Use **cupsaddsmb** para instalar los controladores de *Windows* en *Samba* desde *CUPS*.
4. Conéctese desde un cliente de *Samba* usando un ID que tenga permiso para modificar las configuraciones de impresión en el servidor, y ajuste adecuadamente las características de la impresora predeterminada (tamaño de la página, etc.). Esto es más complejo de lo que parece, ya que *Windows* proporciona dos ventanas idénticas en diferentes partes de la GUI de configuración y sólo una de ellas afecta a los ajustes predeterminados. (Véase el HOWTO de *Samba* para más información).
5. En cada cliente *Windows*, navegue por el entorno de red para encontrar al servidor. Haga clic con el botón derecho sobre la impresora que desee y “conéctela”. La impresora aparece ahora en el grupo de impresoras locales y se puede usar con facilidad.

La instalaciones grandes desearán usar un *script* de conexión para ejecutar el punto 5 en vez de hacerlo a mano.

#### 14.9.5. Imprimir esquemas de migración

En el caso de lugares pequeños con unas cuantas impresoras y estaciones de trabajo, es sencillo establecer un servidor de impresión de tipo GNU/Linux y simplemente reconfigurar a mano cada estación de trabajo de cliente. Si hay varias impresoras compartidas unidas a máquinas escritorio, valdría la pena aprovechar la oportunidad para consolidarlas en un servidor para impresión. Resulta más fácil hacerlo adaptando tarjetas de ethernet a impresoras que admitan la instalación (esto también puede proporcionar una sustancial mejora del funcionamiento con respecto a las interfaces paralelas o de serie). Las impresoras sólo paralelas se pueden preparar para la red usando cajas de impresoras de red.

Ciertamente, los lugares más grandes se beneficiarán del uso de uno o más servidores de impresión. Estas máquinas pueden realizar también otras tareas, pero si hay un importante volumen de impresión debería recordarse que convertir el PostScript a otros formatos es un trabajo fuerte para la CPU y las máquinas habrían de dimensionarse en consonancia con ello. Valdrá la pena establecer la configuración

"Señala e Imprime" si se ha de funcionar con clientes de *Windows*, ya que la migración de máquinas clientes desde los antiguos servidores de impresión *Windows* a los nuevos GNU/Linux se puede hacer con un simple *script* de conexión.

#### **14.9.6. Posibles problemas**

Hay algunos problemas corrientes que se pueden evitar si se ha hecho la oportuna planificación:

1. Asegúrese de que cada impresora es controlada solo por un servidor. Haga que todos los demás escritorios y servidores envíen las tareas de impresión para una impresora a través de su servidor de control. Esto es especialmente importante en el caso de las impresoras unidas a la red. Si no se hace así, la impresora podría recibir dos o más tareas de impresión al mismo tiempo con posibles problemas de salida.
2. Si es posible, intente que sólo un conjunto de controladores formatee la salida de una impresora. Es probable que esto se haga mejor en el servidor de control, pero no necesariamente, ya que depende en cierta medida de qué servidor tenga el mejor controlador para la impresora. Otras máquinas con cola de impresión de la salida debería tratarla como datos en bruto. Si no se hace así, a veces un controlador lo intentará y formateará una serie de salidas ya formateadas, corrompiendo la salida. Es probable que esto sólo sea un problema cuando la salida formateada contenga datos binarios.

#### **14.9.7. Más información sobre la impresión**

En la Web hay mucha información disponible. Estos sitios son útiles para empezar:

<http://cups.org/> - *CUPS*, el Sistema Común de Impresión Unix.

<http://linuxprinting.org/> - el sitio sobre Impresión Linux, con todo un abanico de información útil.

<http://www.linuxprinting.org/kpfeifle/SambaPrintHOWTO> - el HOWTO sobre Impresión de *Samba*. Observe que ésta es sitio de distribución del autor y el documento es un trabajo que se está haciendo. Busque el último borrador.

#### **14.10. Aplicaciones heredadas**

Las aplicaciones que no tienen alternativa OSS y no pueden ser recompiladas para funcionar con OSS habrán de ser ejecutadas en una máquina que haga funcionar el sistema operativo heredado, o con un hardware o emulador de software. Se pueden aplicar las técnicas de las que se habla en este Capítulo.

#### **14.11. Protección frente a los virus**

En los entornos *Windows* y *Macintosh* ahora es fundamental disponer de un paquete antivirus actualizado. Incluso hay entidades con escaso interés en la seguridad que ignoran esta protección y corren un gran riesgo.

Por el contrario, hay muy pocos virus viables que afectan a los sistemas OSS. Como resultado, la protección contra los virus en los entornos OSS suele limitarse a barrer el correo electrónico para evitar que los virus pasen a los usuarios de *Windows*.

En el pasado hubo ataques automatizados a los sistemas OSS, de los que el más famoso fue el 'Gusano Morris'. El fuerte hincapié que se hizo sobre la seguridad desde que sucedió eso redujo los riesgos considerablemente, pero aún es posible que un día se

lance un virus eficaz. Las buenas prácticas en la gestión de los sistemas y la formación continua de los usuario son actualmente mejor defensa que el software antivirus.

En estos momentos se conoce la existencia de dos proyectos antivirus OSS, *Open*

*AntiVirus* (<http://www.openantivirus.org/>) y *ClamAV* (que al parecer se ha evaporado de la red). Los dos están aún muy en mantillas y ninguno es recomendable de momento. Muchos de los productos antivirus comerciales tienen versiones que funcionarán en plataformas OSS. Estas versiones no son totalmente equivalentes a sus pares de *Windows*, pues en estos momentos se destinan a funciones como el barrido del correo electrónico más que a la detección de virus "al vuelo" ejecutando un código como se suele hacer en el entorno *Windows*. Sin embargo, como ya se ha explicado, la detección "al vuelo" no es en absoluto necesaria en los sistemas OSS, pues suele bastar con el barrido del correo electrónico.

#### 14.12. Referencias

<a href="http://www.samba.org/">http://www.samba.org/</a>	El servidor <i>Samba</i> de dominios / impresión/ archivos SMB
<a href="http://www.openldap.org/">http://www.openldap.org/</a>	El servidor de directorios <i>OpenLDAP</i>
<a href="http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html">http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html</a>	La guía del administrador de sistemas Linux-PAM
<a href="http://samba.mirror.ac.uk/samba/docs/man/Samba-HOWTO-Collection.html">http://samba.mirror.ac.uk/samba/docs/man/Samba-HOWTO-Collection.html</a>	La principal colección HOWTO de <i>Samba</i>
<a href="http://www.csn.ul.ie/~airlied/pam_smb">http://www.csn.ul.ie/~airlied/pam_smb</a>	Módulo <i>pam_smb</i> PAM para autenticar a los usuarios GNU/Linux con SMB
<a href="http://samba.idealx.org/index.en.html">http://samba.idealx.org/index.en.html</a>	Herramientas IDEALX y HOWTOs relativos a <i>Samba</i> (algunos en francés).

## 15. Escenario 2 – Unix

### **La Administración tiene servidores Unix ("gran Unix" - Solaris, HP/UX, AIX, OSF/1, etc.). Usa PCs con aplicaciones para servidor/cliente. Algunas Estaciones de Trabajo Unix y Terminales X.**

La migración de los escritorios del PC será similar a la del Escenario 1. Parece que las estaciones de Trabajo y los Terminales X funcionan con aplicaciones de tipo X que deberían ejecutarse sin problemas en los nuevos escritorios. El principal problema consiste en la migración a los servidores.

La migración desde Unix a GNU/Linux es similar a pasar de una versión de Unix a otra. Hay que tener presente que en el término Unix se incluyen los códigos base OSF/1, BSD y AT&T que son diferentes implementaciones del estándar POSIX, como también lo es GNU/Linux. Las diferencias aparecen cuando un programa usa características ajenas a POSIX, como la administración del sistema y las características que mejoran el rendimiento. Los programas mal escritos que se han elaborado sin pensar mucho en el estándar POSIX también tendrán problemas, pues escribir programas transferibles es un arte, y los programas "de andar por casa" van a necesitar algún retoque (la eliminación de todas las advertencias del compilador elaboradas al más alto nivel es un buen comienzo). Pero los problemas son más bien de detalle que de arquitectura fundamental. Unix usa protocolos abiertos como TCP/IP y servicios comunes como DNS y DHCP.

Es probable que también la configuración sea diferente. Aunque no parece que los datos del sistema se vayan a mantener en un formato propietario, por lo que sería bastante fácil tratarlos de modo que se ajusten a los requisitos de GNU/Linux. Esto incluye los nombres de usuario y las contraseñas aunque ciertas sutiles diferencias podrían significar que no es posible una sencilla transferencia.

Si se dispone del código fuente, entonces la recompilación debería trasladar el código. Pero hay algunos temas que habrá que tratar:

1. No hay normas para la ubicación de los archivos y puede ser difícil codificarlos en programas (como `/usr/bin`, `/usr/local/bin` o `/opt/bin`).
2. Hay muchos valores diferentes para las constantes del sistema como, por ejemplo, el número máximo de archivos que se pueden abrir al mismo tiempo.
3. Sutiles diferencias en el lenguaje de programación como, por ejemplo ksh y pdksh. Los diferentes compiladores C son más o menos estrictos respecto al control sintáctico, por lo que el código que se permitiera en una máquina podrían dar un error en otra.

El código podría no ser transferible, por ejemplo, debido a:

- El uso no transferible de constantes, como usar números en lugar de SIGPIPE (algo que se define en un archivo de cabecera C). Este es un ejemplo de un programador que programa para el sistema operativo en lugar de para el estándar POSIX.
- Presunciones sobre la longitud de las palabras o la ordenación de los bytes.

`gcc`, el compilador en GNU/Linux, tiene opciones para ser bastante flexible en estas circunstancias.

4. Cada Unix puede tener diferentes librerías y archivos de encabezados/cabeceras. También pueden estar en distintas ubicaciones. Las ubicaciones y los nombres se pueden cambiar automáticamente una vez que se han encontrado. Pero si la librería o el archivo de

cabeceras presenta un comportamiento diferente, es necesaria una intervención manual. Por ejemplo:

- La semántica de algunas llamadas de librería difieren:
  - como threads
  - exec (setuid bit en *scripts*, ignorado)
  - I/O asíncronos
  - ioctl para control tty
- Diferentes valores para errno.

Puede que el código original use aplicaciones o librerías propietarias que no están disponibles en GNU/Linux. Puede que haya que reescribirlo para usar lo que hay disponible en GNU/Linux. Este podría ser el caso si se necesitan interfaces especiales de hardware como, por ejemplo, una tarjeta para fax. Es posible que haya que trabajar mucho en esas circunstancias.

5. Los makefiles que ayudan a construir aplicaciones pueden tener que ser puestos al día para reflejar las diferencias arriba citadas.
6. Las aplicaciones pueden presuponer cosas sobre subsistemas concretos como bases de datos e impresión, lo que significa, por ejemplo, que el código SQL puede tener que escribirse de nuevo.
7. Llevar cualquier código a un nuevo hardware, compilador o sistema operativo, puede mostrar errores en el programa que siempre estuvieron ahí pero que nunca se pusieron de manifiesto, por ejemplo porque la memoria se empleó de otra manera, los enteros tienen distinto tamaño o los bytes se ordenan de otra forma.

Las siguientes referencias nos dan más información al respecto.

<http://www.linuxhq.com/guides/LPG/node136.html>

[http://www1.ibm.com/servers/esdd/articles/porting\\_linux/index.html?t=gr,l=335,p=PortSolaris2Linux](http://www1.ibm.com/servers/esdd/articles/porting_linux/index.html?t=gr,l=335,p=PortSolaris2Linux)

<http://www-106.ibm.com/developerworks/linux/library/l-solar/?open&t=gr,l=921,p=Sol-Lx>

<http://www->

[ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130115.pdf](http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130115.pdf)

<http://www.unixporting.com/porting-guides.html>

## **16. Escenario 3 - Mainframe**

**La administración se basa en un ordenador mainframe (que podría estar ejecutando MVS, VM/CMS, AS/400, o incluso Unix). La mayoría de los usuarios tienen acceso a las pantallas verdes. Hay algunos PCs, la mayoría utilizados como emuladores en terminal pero con una o dos aplicaciones locales.**

Este Escenario es similar al cliente ligero en lo que respecta al escritorio, especialmente si no se va a cambiar la arquitectura.

No se dispone de datos sobre migración de los servidores

## 17. Escenario 4 – Cliente ligero (*thin client*)

**La Administración usa clientes ligeros con acceso vía Citrix o similar a una mezcla de aplicaciones Windows y Unix.**

El uso del modelo de referencia aquí no se presupone ya que aún se pueden mantener las razones originales para usar un modelo de cliente ligero. Sin embargo, si se contempla un cambio a la Arquitectura de Referencia, entonces aparecerán muchos de los mismos problemas vistos en el Escenario 1. La migración en este Escenario con esta presunción es muy sencilla ya que la Arquitectura no va a cambiar.

Como el cliente es ligero, todo lo que se requiere es un visor OSS para cada protocolo necesario. El sistema de "ventaneo" (*windowing*) no necesita tener mucha funcionalidad, por lo que bastaría con un gestor ligero como *tvwm*.

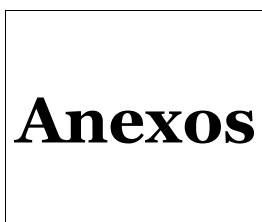
Se pueden admitir los siguientes protocolos, entre otros:

1. **HTTP**. Cualquier navegador OSS bastaría. Habría que investigar la capacidad para ejecutar Javascript y Java. Además, los accesorios necesarios deberían poder funcionar con él directamente a través de un sustituto usando el paquete *plugger* o a través del paquete propietario *CrossOver Plugin* de Code Weavers.
2. **ICA**. Este es el protocolo propietario *Citrix*. *Citrix* proporciona sin cargo un visor ICA que trabaja con GNU/Linux, pero no con OSS.
3. **RDC**. Este es el protocolo usado por el *Windows Terminal Server*. Existe un visor OSS para RDP, el *rdestktop*.
4. **VT220, VT100**, etc. Estos protocolos DEC funcionan todos con *xterm* usando la apropiada configuración de variables de entornos TERM. La conexión al host se hace vía telnet.  
*xterm* puede emular muchos diferentes tipos de terminal cambiando el valor de la variable TERM. Por ejemplo, configurar **TERM=prism9** emulará el protocolo utilizado por las máquinas PRIME. Todas las emulaciones presuponen una conectividad telnet o similar y un protocolo basado en los caracteres más que en las páginas.
5. **3270**. El programa *x3270* presta el apoyo necesario. La conexión al host se hace vía telnet.
6. **X**. Este es el protocolo original de visualización en GNU/Linux y por ello no debería dar problemas.

Hay productos propietarios para algunos de los protocolos más esotéricos.

El Proyecto de Servidor de Terminal Linux (LTSP) <http://www.ltsp.org/> da algunos kits para construir dispositivos para clientes ligeros basado en GNU/Linux. Este es un proyecto extremadamente activo y la calidad del software parece ser muy buena.

Los cambios necesarios en los servidores son similares a los contemplados en las consideraciones de las que se habló en otros Escenarios.



**Anexos**



## Anexo A. Estudios de casos publicados

### A.1. <http://www.turku.fi/tieto/liite44.rtf>

Pruebas de funcionamiento en la Ciudad de Turku de:

1. OpenOffice.org,
2. x3270,
3. Host IBM a petición,
4. WRQ Reflection para la web,
5. Hansa,
6. AS400 Emulator,
7. Netscape Communicator,
8. Mozilla,
9. Konqueror,
10. F-Secure Antivirus.

No se ha informado sobre experiencias de migración real. Intención de emigrar en 2003.

### A.2. <http://www.m-tech.ab.ca/linux-biz>

Una colección de sitios que en apariencia usan GNU/Linux. La información sobre cada sitio es sucinta, pero hay enlaces a posibles contactos que pueden proporcionar más información.

### A.3. <http://www.washingtonpost.com/ac2/wp-dyn/A59197-2002Nov2?language=printer>

Descripción del uso de OSS en Extremadura, España. Se ha creado una distribución local denominada **linex** (véase <http://www.linex.org>). Su experiencia consiste en que los usuarios tienen pocos problemas en usar el software. Además, en algunas circunstancias los usuarios necesitan acceso a *Windows* para trabajar con los formatos de archivo de Microsoft. Recientemente han anunciado que en la actualidad prestan apoyo a 80.000 usuarios.

### A.4. <http://www.newsforge.com/print.pl?sid=02/12/04/2346215>

Es una historia sobre la Ciudad de Largo. Tiene una serie de enlaces y se habla sobre el coste de la propiedad.

### A.5. <http://people.trustcommerce.com/~adam/office.html>

Una compañía que escribió su experiencia de pasarse al OSS en el escritorio así como sus servidores. Un sitio KDE.

### A.6.

#### <http://www.business2.com/articles/mag/print/0,1643,44531,00.html>

Descripción de Zumiez, que puso los escritorios de Ximian Gnome en su entidad.

También otros estudios de casos.

### A.7. <http://lwn.net/Articles/13301/?format=printable>

## **Anexos - Estudios de casos disponibles para el público**

Contiene informes de experiencias en Dinamarca. Se refiere a un informe del Consejo Danés de Tecnología que, y es una pena, no se ha traducido totalmente al inglés.

**A.8.** [http://www.siriusit.co.uk/support/casestudies/k\\_g\\_case.html](http://www.siriusit.co.uk/support/casestudies/k_g_case.html)

Un estudio de casos relativo a una empresa británica que se pasó a OSS recientemente.

**A.9.** <http://staff.harrisonburg.k12.va.us~rlineweaver>

Una experiencia sobre el uso del OSS en los colegios.

**A.10.** <http://www.li.org/success>

Una lista de estudios de casos. Cierta información útil.

**A.11.** <http://www.statskontoret.se/pressrum/press/2003/presso30207english.htm>

<http://www.statskontoret.se/pdf/200308eng.pdf>

<http://www.statskontoret.se/pdf/200308engappendix.pdf>

Un estudio del OSS en la administración pública sueca.

**A.12.**

<http://www.3.ibm.com/software/success/cssdb.nsf/topstoriesFM?OpenForm&Site=linuxatibm>

Una lista de estudios de casos referidos concretamente a IBM. No se dan detalles reales y se centra en el servidor pero da una idea del uso real de GNU/Linux en el mundo.

**A.13.**

<http://h30046.www3.hp.com/search.php?topiccode=linuxCASESTUDY>

Una lista de estudios de casos concretamente referidos a HP.

**A.14.** [http://openapp.biz/seminar/Tony\\_Kenny/Tony\\_Kenny.pdf](http://openapp.biz/seminar/Tony_Kenny/Tony_Kenny.pdf)

En los últimos años el hospital Beaumont en Dublín se ha pasado completamente al OSS no sólo en lo técnico sino también en el aspecto administrativo de la TI. El enlace da bastante detalles. Gracias a se cambio, Beaumont espera ahorrar del orden de 13 millones de Euros en 5 años. Han encontrado que, bien gestionada, la migración resulta aceptable para todo el personal y que los sistemas OSS pueden ejecutarse con mucha mayor eficacia.

## Anexo B. Wine

*Wine* viene de "Wine is Not an Emulator" (*Wine* no es un emulador) y podemos encontrar más información sobre él en <http://www.winehq.com/>.

### B.1. Historia

El desarrollo de *Wine* lo inicio en 1993 Bob Amstadt quien operaba con *GNU/Linux* y *Windows* en la misma máquina. El software *GNU/Linux* había llegado al punto en el que era capaz de satisfacer la mayoría de sus necesidades, pero tenía algunos programas de juegos a los que era muy aficionado que sólo estaban disponibles en *Windows*.

Harto de reiniciar el sistema para ejecutar los juegos, comenzó a trabajar en la manera de interceptar el sistema utilizado para las llamadas de juegos y de asignarlas al entorno *GNU/Linux X*. Otras personas se enteraron de lo que estaba haciendo y se pusieron a ayudarle hasta que *Wine* pudo ejecutar los juegos que les interesaban.

Alrededor de 1995, se intentó ejecutar otros programas como, por ejemplo, *Quicken* y la serie de *Office*, y estas aplicaciones se convirtieron en la principal área de interés. Se formó otro grupo que continuó el apoyo de las técnicas gráficas complejas empleadas en los juegos pero que en general no se utilizan en aplicaciones para oficinas. Entonces el proyecto adquirió un aire más formal al disponer de un equipo de creativos principales y un líder de proyecto. Desde el año 2000 el proyecto ha ido ganando una base más sistemática, con su líder de proyecto y su equipo de apoyo con sede en Estados Unidos, dos pequeños equipos de desarrollo en Canadá y creativos en la mayoría de los países europeos. También los principales vendedores están colaborando, como es el caso de IBM.

Se crearon técnicas para identificar las llamadas al sistema operativo que hizo el programa. En la mayoría de los casos la creación de una pequeña cantidad de código permitiría funcionar a una aplicación dada. Se descubrió que los programas suelen hacer preparaciones para llamar a una interfaz concreta pero entonces no hace la llamada realmente. Entonces se escribió el código para que así los programas pudieran seguir haciendo esas llamadas preparatorias sin incurrir en errores inmediatos, y para que el código apoyara las llamadas reales.

Corel fue el primero en usar *Wine* comercialmente. Corel había trabajado mucho en apoyar a *Wine* y lo utilizó para producir una versión nativa en *GNU/Linux* de *Wordperfect 8*. Desde entonces otras empresas han utilizado a *Wine* para producir una versión *GNU/Linux* de sus productos con el mínimo esfuerzo y cambio, y una de las últimas es Xilinx que produce paquetes electrónicos de CAD para especialistas. El proyecto *Mono* de Ximian parece que va a usar *Wine* para que las aplicaciones .NET creadas por *Windows* funcionen sin volverlas a formular. En <http://appde.winehq.com> se dan más detalles sobre el nivel de apoyo para toda una serie de aplicaciones.

Recientemente, un equipo de creadores de aplicaciones *Windows* experimentados ha empezado a fabricar una serie de programas de tests para comprobar de manera sistemática las más de 12.000 llamadas al sistema presentes en el conjunto de librerías de *Windows*.

Actualmente *Wine* comprende unas 750.000 líneas de código "C" que ponen en marcha más del 90% de las llamadas en de las especificaciones más populares de

## Anexos - Wine

*Windows* como ECMA-234 y Open.32. Las llamadas que no tienen documentación pública son más difíciles de poner en marcha, pero se sigue avanzando.

Algunas empresas trabajan sobre *Wine* para desarrollar código de funciones concretas que inicialmente son propietarias. Lo hacen para financiarse y financiar su trabajo sobre el proyecto principal. Trasladan su código al principal cuando tienen una buena fuente de ingresos alternativa. El apoyo a OLE y ActiveX entra en esta categoría.

### **B.2. Qué hace Wine**

*Wine* intercepta todas las llamadas de los sistemas DOS y *Windows* y junto con BIOS interrumpe y trata de asignarlas al entorno GNU/Linux. Las instrucciones nativas del procesador se ejecutan como si hubieran estado en el entorno *Windows*, y por eso *Wine* no es un emulador completo.

*Wine* no está sujeto a la arquitectura Intel x86; por ejemplo, existe una versión para el DEC/Compaq Alpha, pero una auténtica demanda y un uso importante sólo se da con la x86. Y no permite ejecutar los programas x86 *Windows* en otra arquitectura como la PowerPC o SPARC, si bien *Wine* compilará y ejecutará los dos.

No todas las interfaces del entorno *Windows* se pueden proyectar a una interfaz en el entorno X y *GNU/Linux*. Algunas interfaces *Windows* simplemente no tienen un equivalente. Esto significa que en algunos casos ha de escribirse una significativa cantidad de código para lograr la correspondencia. Por ejemplo, hay problemas con los cursores más complejos usados por algunos programas *Windows*. El Sistema X *Windows* no puede hacer frente a más de dos colores por cursor, lo que significa que *Wine* tiene que suponer qué colores usar, a veces con resultados no utilizables.

*Wine* es realmente dos productos, el propio *Wine* que permite ejecutar los programas *Windows* precompilados, y el *Winelib* que se puede usar para compilar un programa *Windows* de cara a producir un programa nativo GNU/Linux (que es el que empleó Corel para producir la versión GNU/Linux de *Wordperfect*).

*Winelib* podría usarse para ejecutar programas en otro hardware que no sea el x86 si se dispone del código de fuentes, si bien pueden presentarse otros problemas concretos de arquitecturas (por ejemplo, temas de endian).

### **B.3. Los puntos fuertes de Wine**

Existe soporte para los programas *Windows 3.x/95/98/ME/NT* (aunque el apoyo para *Windows NT* es menos completo). Muchos programas ideados para *Windows 2000* podrán ejecutarse a menos que usen nuevos interfaces especializados presentados con *Windows 2000*. Hasta ahora poco trabajo se ha hecho para dar soporte a programas concretos de *Windows XP*, pero algunos hay.

*Wine* da apoyo a la mayoría de los interfaces *Windows* documentados que se conocen si bien el soporte no es siempre tan completo como quisiéramos. Véase <http://winehq.com/?page=status> donde se dan más detalles sobre el estado actual de soporte de *Wine*.

Los programas que se ejecutan aislados o sólo emplean interfaces de comunicaciones externos, normalmente se podrán ejecutar. Cada programa debería ser revisado de manera individual ya que los interfaces y parámetros precisos que se usan podrían interactuar y causar problemas.

Algunas personas han informado sobre compiladores que funcionan y entornos de desarrollo con cierto éxito.

## Anexos - Wine

### **B.4. Los puntos débiles de Wine**

Algunas áreas específicas no está completas, como por ejemplo el *Dynamic Data Exchange* (DDE, Intercambio dinámico de datos). Pero muchos programas hacen llamadas de DDE sin usarlas realmente y por ello funcionarán bastante bien. OpenGL y otras áreas de gráficos de alta velocidad especializados también tienen aspectos a considerar. La puesta en marcha de Listas de Control de Acceso (como en *Windows NT*) ya existen en parte, pero aún no se ha integrado con las ACL en el O/S subyacente.

La tecnología de controlador de dispositivos VxD presentada con *Windows 98* es un área difícil. Necesita acceso al hardware y a los elementos internos de kernel de manera que cualquier sistema serio de multiusuarios podría no permitirla. Se dispone de técnicas para producir equivalentes pero implican un montón de trabajo no hay garantía de que funcionen. En algunos casos se puede convencer al vendedor de que prepare una versión nativa GNU/Linux que use interfaces normales GNU/Linux. Como esta arquitectura está siendo abandonada por Microsoft (las arquitecturas del tipo *Windows NT* no permitirán ese acceso) poco a poco dejará de ser un problema.

Algunos programas *Windows* tratan de manipular directamente los dispositivos (especialmente los puertos de serie). Esto no lo permite GNU/Linux ni otras variantes de Linux. Normalmente sólo se aplica a paquetes de comunicaciones como *Procomm* y a programas derivados de productos DOS donde hayan de hacerse tales cosas.

La reproducción de algunas imágenes gráficas aún no resulta satisfactoria, especialmente el manejo del conjunto de caracteres TrueType. No obstante se está en ello.

Otra área difícil es la constituida por el software creado por Microsoft, ya que estos productos tienden a usar interfaces no documentados. Hasta que se pueda descubrir lo que pasa, los creadores han de tener cuidado pues las leyes sobre ingeniería inversa son muy estrictas en algunos países. Estados Unidos, por ejemplo, prohíbe la ingeniería inversa cualquiera sea su objeto, y la mayoría de los países occidentales sólo la permiten si está destinada a establecer la compatibilidad. Así pues, el trabajo en esta área siempre será algo lento.

Los instaladores de aplicaciones han sido especialmente problemáticos, pero recientemente se han resuelto la mayoría de las dificultades, y se sigue trabajando en este sentido. Algunas de las dificultades se debían a los creadores de paquetes que no usaban las técnicas recomendadas. El acceso al registro es un buen ejemplo. *Wine* mantiene su registro en formato diferente al de *Windows* para facilitar la recuperación. En tanto en cuanto se empleen interfaces documentados para acceder al registro, ello no importa, pero a veces los creadores, aun a riesgo de corromper un auténtico registro de *Windows*, acceden al registro directamente, con el resultado de que no se puede hacer funcionar el programa bajo *Wine*.

A veces se le critica a *Wine* su escaso rendimiento, pero ello suele deberse a un extenso código de búsqueda y corrección de los errores. Se puede compilar *Wine* sin esto, pero habría de hacerse con cuidado pues significa que el problema no se puede diagnosticar sin una nueva compilación.

### **B.5. Wine: alternativas comerciales**

Como ya se ha dicho anteriormente, las versiones extensas de *Wine* están disponibles como productos comerciales para dar soporte a la creación de lo principal de *Wine*.

## Anexos - Wine

Dos empresas que se ocupan de ello son Transgraming y CodeWeavers. Transgraming trabaja principalmente en mejorar los interfaces gráficos y de audio y su producto está pensado para el mercado de los juegos. En CodeWeavers trabajan en las aplicaciones principales de oficina y tienen un producto, *CrossOver Office*, que da soporte, por ejemplo a *Office* y *Lotus Notes*.

### **B.6. Wine y Visual Basic**

De *Visual Basic 3* se ha informado que funciona bajo *Wine*, pero no disponemos de datos concretos.

Actualmente el *Visual Basic 3* no lo instalará. Se está trabajando de cara a hacerlo, pero es muy pronto para decir si será todo un éxito o no.

No se tienen noticias de que se hayan intentado otras versiones.

### **B.7. Aplicación de la migración a Wine**

La siguiente es una relación de las pautas generales a seguir para dirigir el proceso de migración de aplicaciones a GNU/Linux bajo *Wine*:

1. Compruebe las condiciones de la licencia. Algunas empresas han emitido licencias que prohíben ejecutar su aplicación excepto en el sistema operativo final. Oracle, por ejemplo, lo hacía y Microsoft está empezando a hacerlo en el caso de los componentes que se pueden descargar gratis.; la situación es fluida. Elimine de la lista cualquier programa con esas condiciones y haga una nueva lista con ellas.
2. Consiga copias de todas las aplicaciones que han de someterse a migración (que bien pueden ser todas). Puede que las licencias de sitio no permitan sacar copias para pruebas de funcionamiento.
3. Preparar una máquina de pruebas con la última versión de *Wine*.
4. Haga una prueba a todos y cada uno de los programas de la lista de pruebas. Apunte todos los problemas que encuentre, así como si están en la fase de instalación, de inicialización o de ejecución. Valore si realmente afectan a lo que los usuarios necesitan hacer para probar el funcionamiento, y hágalo con una selección representativa de usuarios finales. También se anotará el posible rendimiento escaso. Los mensajes de advertencia se output indicando si las llamadas aún no se han puesto en marcha o si lo han sido de manera incompleta.
5. En el caso de cada programa de la lista de problemas, primero compruebe si ya hay una implementación de GNU/Linux. Si lo hay, no debería haber problemas, pero haga una prueba cuanto antes. Si no existe implementación de GNU/Linux, será necesario ponerse en contacto con el vendedor y sugerirle que cree una a través de *WineLib*. Puede que se trate otra vez de la falta de DLL. Se tratará con cada vendedor por separado.
6. Si los vendedores se niegan a colaborar, habrá que encontrar aplicaciones alternativas o abandonar el proyecto.
7. Una vez que se disponga de la necesaria lista de las DLL de las llamadas a librería, se podrá obtener un precio para la implementación.
8. Habrá que volver a someter a pruebas de funcionamiento cada programa con nuevos análisis selectivos de *WineLib* hasta que los problemas desaparezcan. A veces las correcciones causan problemas a programas que antes funcionaban perfectamente, y esto hay que comprobarlo.

## Anexos - Wine

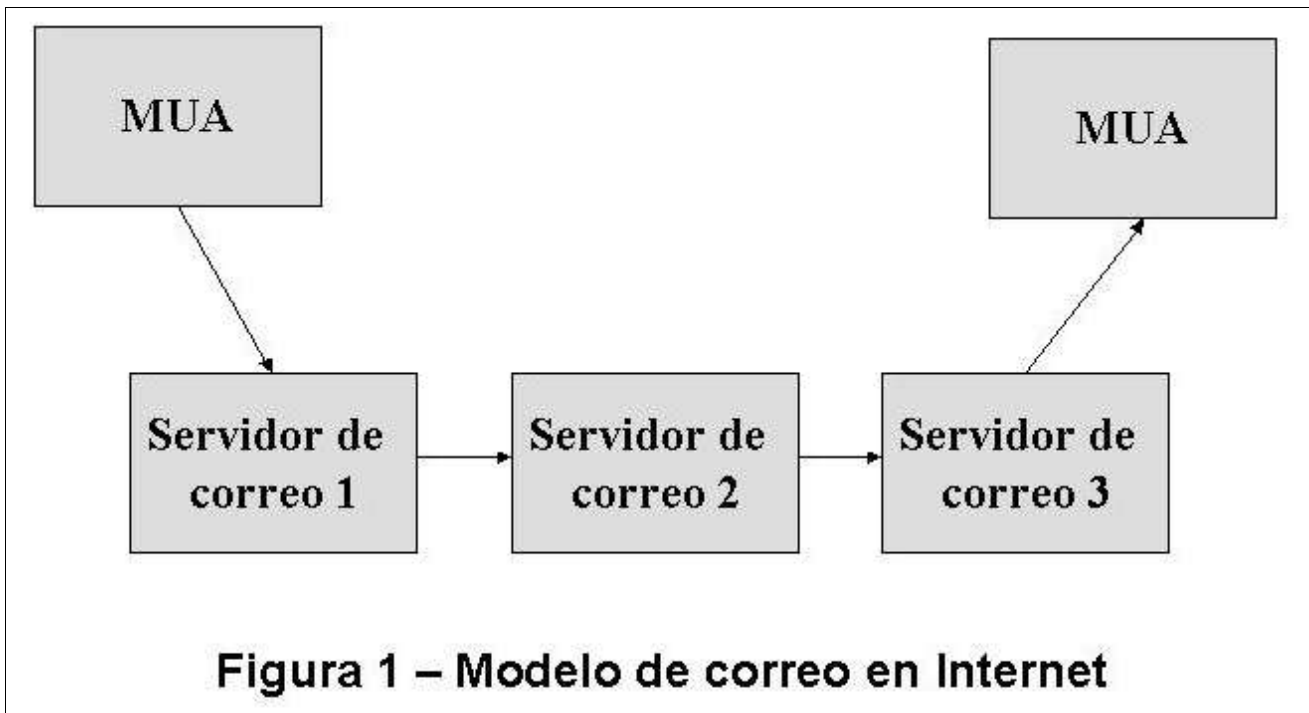
9. Normalmente *Wine* se compila con trazado de búsqueda y corrección de los errores, y esto afecta negativamente a su rendimiento, sobre todo a las interacciones de pantallas. Todo programa que funcione correctamente pero tenga problemas de rendimiento debería volverse a ejecutar con una copia de *Wine* compilada sin el trazado de búsqueda y corrección de errores. Si el rendimiento sigue siendo insatisfactorio, será necesario nuevo trabajo de preparación.

## Anexos - Sistemas de Correo

### Anexo C. Sistemas de Correo

En este Anexo se describen en detalle los sistemas de correo en general, ya que el abanico de productos de correo OSS a veces puede dar lugar a confusiones y la terminología empleada no siempre es clara.

El Modelo de Correo por Internet se basa en un número de componentes lógicos, cada uno de los cuales tiene una tarea concreta que desempeñar y se comunica con los demás a través del uso de sistemas OSS. El modelo se describe mejor con ayuda de algunos diagramas.



En este diagrama vemos el camino a seguir para la entrega de un correo electrónico. Un Agente de Correo de Usuario (*MUA*, *Mail User Agent*) genera el correo y éste pasa a continuación a un servidor de correo que decide si puede entregarlo localmente o ha de pasarlo a otro servidor. El correo va pasando de servidor a servidor hasta que uno de ellos decide que puede entregarlo localmente, y así lo hace. Cuando la entrega ha concluido, el correo ya está disponible para que un MUA lo lea. El MUA final tiene la responsabilidad de recuperar el correo y de pasarlo a una Interfaz de Correo de Usuario (*MUI*, *Mail User Interface*) para que se lo muestre al usuario.

Cómo decide cada servidor de correo si lo entrega localmente o no sería tema suficiente para otro capítulo. En pocas palabras, cada servidor consulta a un archivo de configuración local (o a varios archivos) así como a la información proporcionada por los servidores de DNS (en especial los registros de MX). Y entonces utiliza todo ello para decidir qué se entiende por local. En el caso de correo no local, el servidor utiliza esa información para determinar la dirección del próximo servidor de correo al que enviarle el mismo.

Cada servidor de correo tiene en general la estructura que se ve en la Figura 2.

El Agente de Transporte de Correo (*MTA*, *Mail Transport User*) acepta las conexiones desde otros servidores de correo y MUAs a través del Protocolo Simple de Transferencia de Correo (SMTP). Si el correo no es para entrega local, el MTA lo envía a otro servidor. Si el

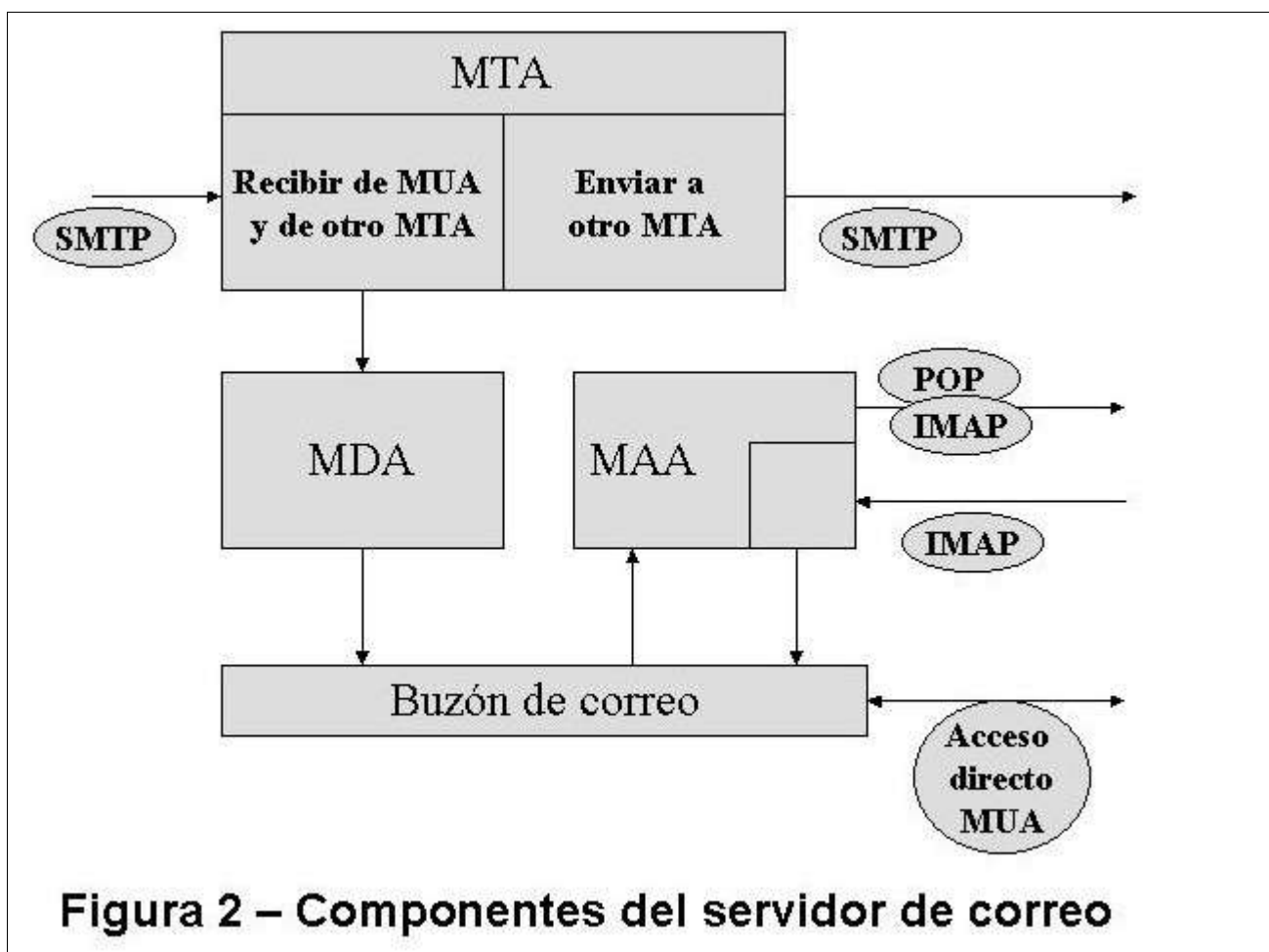


## Anexos - Sistemas de Correo

correo es para envío local, se pasa a un Agente de Entrega de Correo (*MDA, Mail Delivery Agent*). El MDA tiene la responsabilidad de guardar el correo en el buzón del usuario. El buzón es simplemente un modo de guardar datos; por ejemplo un archivo, una serie de archivos independientes o incluso una base de datos de SQL. La estructura concreta de almacenaje es apta para cualquier soporte de MDA.

Cuando un usuario desea ver su correo utiliza un MUA que o bien recupera el correo directamente o bien contacta con la parte componente de servidor que recupera el correo del almacén de correo y se lo traslada al MUA. Estos componentes de servidor no encajan en el modelo tradicional MTA/MDA/MUA y los denominaremos Agentes de Acceso a Correo (*MAA, Mail Access Agents*). Se trata, sin embargo de un término que no es de uso común.

El MUA se comunica con un MAA utilizando un protocolo abierto que habitualmente es o bien *Post Office Protocol (POP)* o bien *Internet Mail Access Protocol (IMAP)*. El protocolo POP normalmente borra el correo del almacén de correo cuando es transferido al cliente y el protocolo IMAP normalmente lo deja como está. El protocolo IMAP también permite que el MUA altere el almacén de correo, por ejemplo borrando correo o moviéndolo de un directorio a otro.



**Figura 2 – Componentes del servidor de correo**

El MUA puede guardar el correo localmente en la máquina que está usando. Esto suele suceder si se usa el POP. Este almacenaje local va a permitir el futuro acceso para ser independiente del servidor, lo que resulta especialmente útil en el caso de máquinas que no están siempre conectadas a la red. El IMAP, por su parte, funciona normalmente sin copias

## Anexos - Sistemas de Correo

locales, pero funciona igualmente en lo que se conoce como modo desconectado, que mantiene una copia local y por ello permite que el correo se “manipule” sin conexión a la red. De este modo, los buzones local y de servidor se sincronizan cuando se hace una conexión a la red. Lamentablemente, no todos los MUA soportan completamente un IMAP desconectado.

A veces, un programa que no es un MUA recupera el correo y lo guarda localmente para que el MUA acceda a él sin tener que conectarse al servidor. Estos programas “atraen” el correo a su máquina en contraste con un MTA al que le llegan los correos “empujados” por otros MTAs. Esto puede resultar útil si los usuarios no quieren permitir las conexiones a sus máquinas desde Internet, o si trabajan tras un cortafuegos. Un ejemplo de este tipo de programas es el *fetchmail*.

La dificultad con este modelo reside en que las aplicaciones disponibles no siempre mapean directamente en él. Con mucha frecuencia, las aplicaciones hacen más de una de las funciones; un MTA, por ejemplo, puede incorporar la función de MDA, y en algunas circunstancias el popular MTA *Sendmail* se puede usar incluso como un MUA.

Cuando un correo pasa a través de varios servidores desde del MUA que lo origina hasta el MUA final, añadiéndose una serie de encabezamientos que registran los detalles del recorrido y controlan el proceso del correo mediante los servidores que intervienen y el MUA final. Algunos de ellos son encabezamientos MIME (*Multi-purpose Internet Mail Extensión* o Extensión para Correo Internet Polivalente) que se usan para muy diversos fines como, por ejemplo, soporte para conjuntos de caracteres no ASCII, contenidos intercalados como las imágenes y para elementos adjuntos. Cuando un MUA adjunta un archivo registra su tipo como encabezamiento MIME y entonces la responsabilidad de poder descodificarlo recae en el MUA.

A continuación se habla más a fondo de este modelo y las partes que lo integran.

### C.1. MTA

La mayoría de los MTA permiten que el administrador controle si un correo se ha de aceptar o no según su procedencia. Esto se hace limitando la gama de direcciones IP desde las que el MTA aceptará conexiones. Esto es de gran valor para evitar que los que envían *spam lo hagan* usando el MTA como relé (repetidor) y atasquen el ancho de banda de la red para el MTA.

Hay una conjunto de más o menos 20 extensiones para el SMTP llamado *Extended SMTP* o ESMTP. Estas extensiones permiten, entre otras cosas, una transferencia de correo más rápida entre los MTA que se acomodan a ello, usando la extensión de redireccionamiento.

Otra extensión permite el cifrado con Seguridad de Capa de Transporte (TLS) entre los MTA que lo permiten y otro, SMTP-AUTH, que permite la autenticación de los usuarios a través de una serie de técnicas. Las dos extensiones son útiles cuando el MTA no permitiría normalmente a un cliente que se conectara porque su dirección IP queda fuera de su espacio de direcciones contrastadas. Esto sucedería, por ejemplo, si un usuario de laptop ingresa por línea telefónica desde un sitio aleatorio en Internet. Véase el punto C.4.2.

El modelo original daba por supuesto que el titular de una cuenta de correo tenía una cuenta de conexión en el servidor de correo. Esto quiere decir que el MTA podría consultar el archivo local de contraseñas para autenticar a los usuarios. Este modelo es demasiado restrictivo y los MTA modernos ahora ya pueden soportar Usuarios Virtuales donde los datos del titular de la cuenta se mantienen en una

## Anexos - Sistemas de Correo

base de datos, con frecuencia independientemente de los datos de a cuenta de conexión normal. Esto significa que un usuario podría tener una contraseña para el correo y otra diferente para conectarse.

La base de datos podría ser de las que funcionan con LDAP, una de tipo SQL o un archivo plano. *MySQL* es el servidor de SQL preferido ya que resulta eficaz y rápido en lo que es básicamente una aplicación de sólo lectura. *PostgreSQL* y *Oracle* también serían adecuados.

Se recomienda una base de datos de tipo LDAP ya que ofrecen mejor soporte a la distribución.

Las implantaciones de LDAP predeterminado suelen usar productos de Bases de Datos Berkeley de Sleepycat Systems.

A veces una máquina sólo puede conectarse a un servidor de correo de manera intermitente. Esto puede darse en el caso de trabajadores en casa o usuarios de laptops, por ejemplo. También puede darse en oficinas pequeñas donde el coste de una conexión permanente no se puede justificar. En estas circunstancias, el MTA central no puede enviar correo como lo haría normalmente y necesita guardarlo hasta que se haga una conexión. Lo mismo vale para el MTA (cuando lo hay) en la máquina del cliente o, en el caso de una oficina pequeña, el servidor de correo de puerta de enlace. Estos MTA necesitan poder soportar esas situaciones y a veces se los denomina Smart Hosts cuando lo hacen.

La entrega desde un Smart Host se puede realizar usando SMTP o POP3.

La entrega a través del SMTP es sencilla y la seguridad en la máquina receptora se puede mejorar restringiendo la conexión interior sólo desde el Smart Host.

La entrega a través de POP3 se lleva a cabo usando un MUA o la aplicación *fetchmail*. *Fetchmail* descargará el correo a un buzón local tal y como se ha dicho anteriormente o lo entregará a un MTA local en caso necesario, por ejemplo cuando hay implicadas múltiples cuentas de correo.

Estos dos métodos funcionan bien pero tienen la desventaja de que no permiten el uso de listas negras para evitar que el spam abra redireccionamientos y se acepten otras fuentes indeseables. Herramientas como *SpamAssassin* pueden eliminar la mayoría del spam, pero los costes de este proceso son mucho mayores, y se usa un mayor ancho de banda para bajar el correo para ser examinado.

### **C.2. MUA**

El MUAS y el MUI juntos forman un paquete que la mayoría de los usuarios consideran “el gestor de correo”. Es el software de cliente que funciona tanto en un servidor de web como directamente en una máquina con escritorio para que la gente envíe y reciba correo. Se suele facilitar algún tipo de almacenamiento para que se pueda archivar el correo en “carpetas” o “buzones locales” para futura referencia.

El MUA maneja protocolos como SMTP para la presentación de correo y POP o IMAP para recuperación y archivo. Entiende el formato de los mensajes de correo y puede descomponer mensajes de MIME en las diversas partes que los componen.

## Anexos - Sistemas de Correo

Cuando se necesita una mayor seguridad de punta a punta, el MUA también es responsable de cifrar y firmar los mensajes. Hay dos estándares que compiten en esto: S/MIME, que se basa en certificados X.509, y PGP/GPG que se basa en un formato de certificado diferente con un modelo fiable para la red en lugar de uno basado en la jerarquía.

La mayoría de los MUA OSS funcionan con firmas digitales usando el *GNU Privacy Guard (GPG)*. Pocos funcionan con firmas S/MIME. Las entidades empresariales y gubernativas han optado por el estándar S/MIME y se debe apoyar su uso.

## Anexos - Sistemas de Correo

### C.3. Buzón

En un principio, los sistemas de correo Unix daban por sentado que el propietario de una cuenta de correo tenía acceso a la máquina que alberga al servidor de correo y podía leer un archivo que contuviera su correo o bien, como alternativa, ese correo podía entregarse a la máquina que el usuario solía conectar para trabajar. Esto estaba bien en el caso de entornos con pocos usuarios que necesitaban una cuenta con conexión real en una máquina con servidor de correo, pero no siempre es practicable ni seguro.

El formato original para guardar el correo era un sencillo archivo por usuario, en el que el correo se iba añadiendo al final. Este archivo podía llegar a ser muy grande e ir “repasándolo” para leer una pieza de correo pronto se mostró ineficaz. Este formato suele conocerse como “mbox” y algunos MUA lo siguen usando, en especial en el caso del correo guardado localmente por el usuario. Una modificación a esto consistía en mantener un correo en un archivo separado dentro de una estructura de directorio, lo que permite un acceso aleatorio más eficaz. Una variante de esta estructura es la denominada “mh”, y otra con ciertos subdirectorios y procedimientos de acceso definidos se denomina “maildir”.

A veces estas estructuras se mantenían en el servidor de correo y se exportaban a los clientes usando, por ejemplo, NFS, lo que permitía mantener el correo centralmente. Esto significa que se podría hacer una copia de seguridad adecuada, aunque aparecían problemas de bloqueo con la estructura de archivo único. Aun así, usar un NFS no se ha generalizado, quizás por la falta de buenos clientes de NFS en *Windows*.

No todos los MTA funcionan directamente con estos diferentes métodos de acceso, de ahí la necesidad de MAAs. Un MUA que no puede acceder al buzón de correo directamente tendrá que usar un componente MAA que se sirva de POP o IMAP.

Tanto el POP como el IMAP envían contraseñas como texto completo predeterminado. IMAP puede usar contraseñas hash si el MUA funciona con ellas. Usar enlaces cifrados de TLS se puede hacer si el MAA y el MUA funcionan con ellos, resulta aconsejable en las redes locales y debería ser obligatorio en el caso de acceso remoto.

A veces los MTA se comunican con los MDA usando el Protocolo de Transferencia de correo Local (LMTP). La mayoría de los MTA y los MDA funcionan con este protocolo.

### C.4 Usuarios roaming

El problema con los usuarios *roaming* es que no pueden conectarse desde direcciones IP impredecibles, por lo que los métodos normales usados por los MTA para decidir si aceptar el correo entrante les evitará el enviar correo a través del servidor de correo de la Administración. Los MTA tienen que restringir el acceso a ellos desde clientes desconocidos para evitar su uso por parte de los spammers como redirección a terceros.

Hay tres técnicas generales para sortear este problema.

#### C.4.1. Redes Privadas Virtuales (VPNs)

En una VPN, a la máquina remota se le puede asignar una dirección que se incluya en un espacio para direcciones confrontadas de los MTA. El problema es que el pleno acceso a la red interna será posible para alguien que

## Anexos - Sistemas de Correo

acceda a la máquina remota, un riesgo importante para los laptops a menos que las claves de acceso estén cifradas con una contraseña introducida cada vez que se inicia la conexión. Lamentablemente, a veces los usuarios configuran sus máquinas para recordar las contraseñas.

### C.4.2. MTP-AUTH y TLS.

La extensión de SMTP-AUTH al SMTP permite configurar el MTA para que pida una contraseña para autenticar al usuario remoto. Los principales métodos de autenticación son PLAIN, LOGIN y CRAM-MDS.

PLAIN requiere que la contraseña esté clara para el cliente pero se pueda cifrar de cara al servidor. Si la conexión SMTP está cifrada entonces la contraseña se pone completa (aunque en tipo 64) por la red.

LOGIN es menos eficaz que PLAIN ya que necesita tres interacciones de red en lugar de una y, como PLAIN, el nombre de usuario y la contraseña viajan completos por la red.

CRAM-MDS cifra el nombre de usuario y la contraseña cuando pasan por la red. Sin embargo, se debe tener la contraseña en texto completo tanto en el usuario como en el servidor. Sólo requiere dos interacciones en red.

No todos los MUA funcionan con SMTP-AUTH y los que lo hacen sólo pueden soportar un número limitado de métodos. *Outlook Express*, por ejemplo, sólo usa LOGIN.

Si se compara con usar una VPN, el único acceso permitido es enviar correo, por lo que otros servicios no se verían comprometidos si la máquina remota resultara robada.

ESMTP también permite negociar una sesión TLS entre el cliente y el servidor. Esta conexión cifra los datos en la red y también puede autenticar la máquina cliente. La autenticación requiere un certificado de cliente que concuerde con otro que haya en el servidor.

### C.4.3. POP-antes de-SMTP.

Este método se beneficia del hecho de que los protocolos POP e IMAP necesitan autenticación de contraseña.

Tras una satisfactoria conexión a POP o IMAP para leer el correo, el MAA mantiene una base de datos de conexiones autenticadas con dirección de PI, la fecha y la hora del cliente. Cuando el cliente trata de mandar correo que no es para el dominio local, el MTA comprueba si la dirección IP del cliente está en su espacio para direcciones de confianza. Si no es así, entonces comprueba la base de datos de logias autenticados para la dirección de PI. Si no se ha registrado ninguna conexión autenticada desde la dirección IP del cliente, o la última conexión autenticada no se ha producido lo bastante recientemente, el MTA se niega a redireccionar el mensaje. El período de tiempo es configurable y suele preestablecerse en 20 minutos. Este método necesita que el MAA y el MTA cooperen. Esta es la razón por la que no todas las combinaciones funcionan.

Este método tiene el inconveniente de que los usuarios deben comprobar el correo de entrada antes. Para algunos usuarios esto puede ser difícil a menos que el MTA lo haga automáticamente en su lugar.

## Anexos - Sistemas de Correo

### C.5. Rendimiento

En general, un MTA usa poca potencia del procesador; los hosts que no ejecutan nada más suelen verse limitados por el ancho de banda de la red o el desempeño del disco. Los servidores IMAP y POP requieren más potencia de procesador y el IMAP un poco más de RAM que el POP. Sin embargo, no parece que ninguno de ellos sea un problema sobre el hardware actual.

Los *scanners* antivirus exigen mucha RAM y energía de procesador, especialmente si se permiten elementos adjuntos de MIME.

Aun así, los límites del rendimiento los suele establecer el tráfico más que el número de cuentas.

**Sitio 1** – 2 x Pentium III Xeon 2.4G, 4Gb RAM, 3 x 36GB SCSI Raid 5

Usuarios Virtuales con la mirada puesta en MySQL.

*Postfix 2.0.6., Courier-IMAP 1.7, MySQL 4.12, RAV-Antivirus, Mailman 2.1, Red Hat Linux 8.0, no SSL.*

Unos 4.800 usuarios.

**Sitio 2** – Athlon 1200, 1 Gb RAM, Raid 5

*Postfix + Courier-IMAP* (escaneo de antivirus en otra máquina) no SSL.

8.500 usuarios.

**Sitio 3** – Pentium 133, 40 Mb RAM, disco IDE.

*Debian GNU/Linux, Courier-MTA + Courier-IMAP + SpamAssassin* (el último sólo para un usuario).

Normalmente 18 usuarios de OPO3 y 7 de IMAP en cualquier momento.

Procesador ocupado en un 20%.

**Sitio 4** - Pentium II 450 Xeon dual, 256 Mb RAM

*MySQL, Courier-MTAP, Courier-IMAP, sqwebmail, SSL.*

50 usuarios, sobre todo POP3.

**Sitio 5** – Pentium II 400 con 256M RAM

*Courier-MTA + SpamAssassin, Red Hat Linux 8.0*

300 buzones, unos 4.000 mensajes al día.

**Sitio 6** – Pentium III 677 Mhz, 512 Mb RAM, 2 discos IDE

*FreeBSD 4.7, Exim 4.05, OpenLDAP 2.1.5., Cyrus 2.1.11, Mailman 2.1, Apache 1.3.26.*

La máquina es sobre todo un servidor de web ocupado, pero también maneja varios miles de correos al día sin ninguna carga adicional apreciable.

# Anexos - Software de Referencia para el Escritorio

## Anexo D. Software de Referencia para el Escritorio

En la siguiente lista aparecen los paquetes de RPM y los números de sus versiones. Esta lista se basa en *Red Hat Linux 8.0* y en el software de actualización *Red Carpet* de Ximian. Sólo incluye un conjunto mínimo de dependencias para el entorno final. Se ha actualizado *Evolution* a través de *Red Carpet* para una versión posterior a la entregada con *Red Hat*.

<b>NOMBRE</b>	<b>VERSIÓN</b>	<b>REVISIÓN</b>
Anacron	2.3	23
Aspell	0.33.7.1	ximian.4
Atk	1.0.3	1
Audiofile	0.2.3	3
Basesystem	8	1
bash	2.05b	5
bdf flush	1.5	21
bitmap-fonts	0.2	2
bonobo	1.0.21	1.ximian.1
bonobo-activation	1.0.3	2
bonobo-conf	0.16	1.ximian.1
bzip2-libs	1.0.2	5
chkconfig	01/03/06	3
chkfontpath	01/09/06	3
compat-libstdc++	7.3	2.96.110
control-center	2.0.1	8
cpp	3.2	7
cracklib	2.7	18
cracklib-dicts	2.7	18
crontabs	1.1	4
cups-libs	01/01/17	0.2
cyrus-sasl	02/01/10	1
cyrus-sasl-md5	02/01/10	1
db4	4.0.14	14
desktop-backgrounds-basic	2	10
desktop-backgrounds-extra	2	10
desktop-file-utils	0.3	3
dev	03/03/01	2
dhclient	3.opl1	26
dialog	0.9b	20020519.1
diffutils	02/08/01	3
docbook-dtds	1	14
e2fsprogs	1.27	9
ed	0.2	28
eel2	2.0.6	1
emacs	21.2	18
eog	1.0.2	5
esound	0.2.28	1
evolution	01/02/02	1.ximian.1
expat	1.95.4	1
fam	02/06/08	4



## Anexos - Software de Referencia para el Escritorio

<i>NOMBRE</i>	<i>VERSIÓN</i>	<i>REVISIÓN</i>
filesystem	02/01/06	5
fileutils	04/01/09	11
findutils	04/01/07	7
fontconfig	2	3
fortune-mod	1	24
freetype	02/01/02	7
gail	0.17	2
galeon	01/02/06	0.8.0
gawk	03/01/01	4
Gconf	1.0.9	6
Gconf2	01/02/01	3
gdbm	01/08/00	18
gdk-pixbuf	0.18.0	4
gdk-pixbuf-gnome	0.18.0	4
gdm	2.4.0.7	13
gedit	2.0.2	5
gftp	2.0.13	5
ghostscript	7.05	20
ghostscript-fonts	5.5	7
gimp	01/02/03	9
gimp-print	04/02/01	5
glib	01/02/10	8
glib2	2.0.6	2
glibc	02/03/02	4.80.6
glibc-common	02/03/02	4.80.6
Glide3	20010520	19
gmp	4.1	4
gnome-desktop	2.0.6	4
gnome-libs	1.4.1.2.90	22
gnome-mime-data	2.0.0	9
gnome-panel	2.0.6	9
gnome-session	2.0.5	7
gnome-spell	0.5	1.ximian.3
gnome-terminal	2.0.1	5
gnome-utils	2.0.2	5
gnome-vfs	1.0.5	6.ximian.1
gnome-vfs2	2.0.2	5
gnupg	1.0.7	6
gpm	1.19.3	23
grep	02/05/01	4
grub	0.92	7
gtk+	01/02/10	22
gtk2	2.0.6	8
gtkhtml1.1	01/01/08	1.ximian.1
gzip	01/03/03	5
htmlview	2.0.0	6
hwdata	0.48	1
imlib	01/09/13	9
indexhtml	8	1

## Anexos - Software de Referencia para el Escritorio

<i>NOMBRE</i>	<i>VERSIÓN</i>	<i>REVISIÓN</i>
info	4.2	5
initscripts	6.95	1
intltool	0.22	3
iproute	02/04/07	5
iputils	20020124	8
kbd	1.06	26
krb5-libs	01/02/05	15
krbafs	01/01/01	6
less	358	28
libacl	2.0.11	2
libart_lgpl	02/03/10	1
libattr	2.0.8	3
libbonobo	2.0.0	4
libbonobo-confo	0.16	1.ximian.1
libbonoboui	2.0.1	2
libcapplet0	1.4.0.1	9
libelf	0.8.2	2
libgal21	0.23	1.80.ximian.1
libgcc	3.2	7
libghttp	1.0.9	5
libglade	0.17	8
libglade2	2.0.0	2
libgnome	2.0.2	5
libgnomecanvas	2.0.2	1
libgnomeprint	1.116.0	2
libgnomeprint15	0.37	2.ximian.1
libgnomeprintui	1.116.0	1
libgnomeui	2.0.3	3
libgtkhtml1.1-3	01/01/08	1.ximian.1
libjpeg	6b	21
libmng	1.0.4	1
libpng10	1.0.13	6
libpng	01/02/02	8
librpm404	4.0.4	8x.27
librsvg2	2.0.1	1
libstdc++	3.2	7
libtermcap	2.0.8	31
libtiff	03/05/07	7
libungif	04/01/00	13
libuser	0.51.1	2
libwnck	0.17	1
libxml	01/08/17	5
libxml2	02/04/23	1
libxslt	1.0.19	1
linc	0.5.2	2
logrotate	03/06/05	2
losetup	2.11r	10
lvm	1.0.3	9
metacity	2.4.0.92	5

## Anexos - Software de Referencia para el Escritorio

<i>NOMBRE</i>	<i>VERSIÓN</i>	<i>REVISIÓN</i>
mingetty	1	3
mkinitrd	03/04/28	1
mktemp	1.5	16
modutils	02/04/18	2
mount	2.11r	10
mozilla	1.0.1	26
mozilla-nspr	1.0.1	26
mozilla-nss	1.0.1	26
mozilla-psm	1.0.1	26
nautilus	2.0.6	6
ncurses	5.2	28
net-tools	1.6	7
newt	0.51.0	1
nfs-utils	1.0.1	2
nscd	02/03/02	4.80.6
nss_ldap	198	3
oaf	0.6.10	1.ximian.2
Omni	0.7.0	6
openjade	01/03/01	9
openldap	2.0.27	02/08/00
openoffice	1.0.1	8
openoffice-i18n	1.0.1	8
openoffice-libs	1.0.1	8
openssh	3.4p1	2
openssh-clients	3.4p1	2
openssh-server	3.4p1	2
openssl	0.9.6b	33
ORBit	0.5.13	5
ORBit2	02/04/01	1
pam	0.75	46.8.0
pango	01/01/01	1
passwd	0.67	3
patch	02/05/04	14
pcre	3.9	5
perl	05/08/00	55
perl-Filter	1.28	9
popt	1.7	1.06
portmap	4	46
procps	2.0.7	25
psmisc	20.2	6
pspell	0.12.2	ximian.7
python	02/02/01	17
qt	3.0.5	17
rc	01/02/01	1.ximian.1
rcd	01/02/01	1.ximian.3
readline	4.3	3
redhat-artwork	0.47	3
redhat-logos	01/01/06	2
redhat-menus	0.26	1

## Anexos - Software de Referencia para el Escritorio

<i>NOMBRE</i>	<i>VERSIÓN</i>	<i>REVISIÓN</i>
redhat-release	8	8
rootfiles	7.2	4
rpm	4.1	1.06
scrollkeeper	0.3.10	7
sed	3.02	13
setup	02/05/20	1
sgml-common	0.6.3	12
shadow-utils	20000902	12.8
sh-utils	2.0.12	3
slang	01/04/05	11
soup	0.7.11	1.ximian.1
switchdesk	03/09/08	9
syslogd	01/04/01	10
SysVinit	2.84	5
tar	1.13.25	8
termcap	11.0.1	13
tetex	1.0.7	57.1
tetex-fonts	1.0.7	57
textutils	2.0.21	5
tmpwatch	02/08/04	3
urw-fonts	2	26
usermode	1.63	1
utempter	0.5.2	10
util-linux	2.11r	10
Vflib2	2.25.6	8
vixie-cron	3.0.1	69
vnc	3.3.3r2	39.2
vnc-doc	3.3.3r2	39.2
vte	0.8.19	2
which	2.14	1
words	2	20
Xaw3d	1.5	16
Xfree86	04/02/00	72
Xfree86-75dpi-fonts	04/02/00	72
Xfree86-base-fonts	04/02/00	72
Xfree86-font-utils	04/02/00	72
Xfree86-libs	04/02/00	72
Xfree86-Mesa-libGL	04/02/00	72
Xfree86-Mesa-libGLU	04/02/00	72
Xfree86-truetype-fonts	04/02/00	72
Xfree86-xauth	04/02/00	72
Xfree86-xfs	04/02/00	72
Xft	2	1
xinetd	02/03/07	5
xinitrc	3.31	1
xml-common	0.6.3	12
xpdf	1.01	10
xscreensaver	4.05	6
xsri	02/01/00	3

## Anexos - Software de Referencia para el Escritorio

<i>NOMBRE</i>	<i>VERSIÓN</i>	<i>REVISIÓN</i>
zlib	01/01/04	8.8x

# Anexos - Software de Referencia para el Servidor

## Anexo E. Software de Referencia para el Servidor

En la siguiente lista aparecen los paquetes de RPM y los números de sus versiones. Esta lista se basa en *Red Hat Linux 8.0* y en el software de actualización *Red Carpet* de Ximian. Cabe observar que no se trata de una lista mínima. Aquí hay aplicaciones que se dejaron por comodidad, como por ejemplo las relativas a *X* y a *Gnome*. Estas permiten el uso de interfaces gráficas, pero también se pueden utilizar las sencillas interfaces de tipo texto.

NOMBRE	VERSIÓN	REVISIÓN
a2ps	4.13b	24
acl	2.0.11	2
adjtimex	1.13	4
alchemist	1.0.24	4
amanda	2.4.2p2	9
anaconda-help	8	1
anacron	2.3	23
apmd	3.0.2	12
ash	0.3.8	5
at	03/01/08	31
atk	1.0.3	1
attr	2.0.8	3
audiofile	0.2.3	3
authconfig	04/02/12	3
autoconvert	0.3.7	8
autofs	03/01/07	33
autorun	3.3	3
basesystem	8	1
bash	2.05b	5.1
bash-doc	2.05b	5.1
bdf flush	1.5	21
beecrypt	02/02/00	6
bg5ps	01/03/00	9
bind	09/02/01	9
bind-utils	09/02/01	9
bitmap-fonts	0.2	2
blas-man	3	18
bonobo-activation	1.0.3	2
booty	0.12	1
bridge-utils	0.9.3	6
bzip2	1.0.2	5
bzip2-libs	1.0.2	5
caching-nameserver	7.2	4
cadaver	0.20.5	2
cdrdao	01/01/05	10
chkconfig	01/03/06	3
chkfontpath	01/09/06	3
cleanfeed	0.95.7b	17
comps-extras	8	3
courier-imap	01/07/01	2.whoson.8.0

## Anexos - Software de Referencia para el Servidor

NOMBRE	VERSIÓN	REVISIÓN
courier-imap-ldap	01/07/01	2.whoson.8.0
cpio	02/04/02	28
cracklib	2.7	18
cracklib-dicts	2.7	18
crontabs	1.1	4
cups-libs	01/01/17	0.7
curl	07/09/08	1
cWnn-common	1.11	27
cyrus-sasl	02/01/10	1
cyrus-sasl-gssapi	02/01/10	1
cyrus-sasl-md5	02/01/10	1
cyrus-sasl-plain	02/01/10	1
db4	4.0.14	14
db4-java	4.0.14	14
dev	03/03/01	2
dhclient	3.0pl1	26
dhcp	3.0pl1	26
dialog	0.9b	20020519.1
dictd	01/05/05	3
diffutils	02/08/01	3
diskcheck	1.3	2
docbook-dtds	1	14
dos2unix	3.1	12
dosfstools	2.8	3
dtach	0.5	5
e2fsprogs	1.27	9
ed	0.2	28
eject	2.0.12	7
elinks	0.3.2	1
enscript	01/06/01	22
esound	0.2.28	1
ethtool	1.6	2
exim	4.12	4.rh8x
exim-ldap	4.12	4.rh8x
expat	1.95.4	1
fam	02/06/08	4
fbset	2.1	11
fetchmail	05/09/00	21/08/00
file	3.39	9
filesystem	02/01/06	5
fileutils	04/01/09	11
findutils	04/01/07	7
finger	0.17	14
fontconfig	2	3
freetype	02/01/02	7
FreeWnn-common	1.11	27
FreeWnn-libs	1.11	27
ftp	0.17	15
ftpcopy	0.5.1	1

## Anexos - Software de Referencia para el Servidor

NOMBRE	VERSIÓN	REVISIÓN
gail	0.17	2
gawk	03/01/01	4
GConf2	01/02/01	3
gd	01/08/04	9
gdbm	01/08/00	18
genromfs	0.3	12
glib	01/02/10	8
glib2	2.0.6	2
glibc	02/03/02	4.80.6
glibc-common	02/03/02	4.80.6
gmp	4.1	4
gnome-audio-extra	01/04/00	4
gnome-mime-data	2.0.0	9
gnome-python2	1.99.11	8
gnome-python2-bonobo	1.99.11	8
gnome-python2-canvas	1.99.11	8
gnome-python2-gtkhtml2	1.99.11	8
gnome-vfs2	2.0.2	5
gnupg	1.0.7	8
gpg-pubkey	db42a60e	37ea5438
gpm	1.19.3	23
grep	02/05/01	4
groff	1.18	6
groff-perl	1.18	6
grub	0.92	7
gsl	01/01/01	3
gtk+	01/02/10	22
gtk2	2.0.6	8
gtkhtml2	2.0.1	2
gzip	01/03/03	5
h2ps	2.06	6
hdparm	5.2	1
hesiod	3.0.2	21
hotplug	2002_04_01	13
htmlview	2.0.0	6
httpd	2.0.40	11.5
hwdata	0.48	1
imlib	01/09/13	9
indexhtml	8	1
inews	02/03/03	5
info	4.2	5
initscripts	6.95	1
intltool	0.22	3
iproute	02/04/07	5
iptables	1.2.6a	2
iptraf	02/07/00	3
iputils	20020124	8
ipxutils	2.2.0.18	11
isicom	3.05	6



## Anexos - Software de Referencia para el Servidor

NOMBRE	VERSIÓN	REVISIÓN
jcode.pl	2.13	6
jfsutils	1.0.17	3
kakasi	02/03/04	8
kakasi-dict	02/03/04	8
kbd	1.06	26
kbdconfig	01/09/16	1
kcc	2.3	14
kdoc	3.0.0	2.cvs20020321.3
kernel	02/04/20	13.8
kernel	02/04/20	18.8
kernel-utils	2.4	8.28
krb5-libs	01/02/05	15
krbafs	01/01/01	6
krbafs-utils	01/01/01	6
ksymloops	02/04/05	1
kudzu	0.99.69	1
lapack-man	3	18
less	358	28
lftp	02/05/02	5
libacl	2.0.11	2
libaio	0.3.13	5
libao	0.8.3	1
libart_lgpl	02/03/10	1
libattr	2.0.8	3
libbonobo	2.0.0	4
libbonoboui	2.0.1	2
libcap	1.1	12
libelf	0.8.2	2
libesmtp	0.8.12	2
libf2c	3.2	7
libgcc	3.2	7
libghttp	1.0.9	5
libglade2	2.0.0	2
libgnome	2.0.2	5
libgnomecanvas	2.0.2	1
libgnomeui	2.0.3	3
libgtop	1.0.12	11
libIDL	0.8.0	3
libjpeg	6b	21
libmng	1.0.4	1
libobjc	3.2	7
libogg	1	1
libole2	0.2.4	4
libpng10	1.0.13	6
libpng	01/02/02	8
librpm404	4.0.4	8x.27
libstdc++	3.2	7
libtermcap	2.0.8	31
libtiff	03/05/07	7

## Anexos - Software de Referencia para el Servidor

NOMBRE	VERSIÓN	REVISIÓN
libtool-libs	01/04/02	12
libungif	04/01/00	13
libunicode	0.4	9
libusb	0.1.6	1
libuser	0.51.1	2
libwvstreams	3.7	5
libxml10	1.0.0	11
libxml2	02/04/23	1
libxml2-python	02/04/23	1
libxslt	1.0.19	1
lilo	21/04/04	20
linc	0.5.2	2
lm_sensors	02/06/03	2
lockdev	1.0.0	20
logrotate	03/06/05	2
logwatch	2.6	8
lokkit	0.5	21/08/00
losetup	2.11r	10
LPRng	03/08/09	6.1
lvm	1.0.3	9
m2crypto	0.05_snap4	6
m4	01/04/01	11
macutils	2.0b3	22
mailcap	02/01/12	1
mailman	2.0.13	3
mailx	08/01/01	26
make	3.79.1	14
man	1.5k	0.8x.0
man-pages	1.53	1
mc	04/05/55	12
mdadm	1.0.0	6
mew-common	2.2	6
mingetty	1	3
mkbootdisk	01/04/08	1
mkinitrd	03/04/28	1
mkisofs	1.1	14
mktemp	1.5	16
mod_perl	1.99_05	3
mod_python	3.0.0	10
modutils	02/04/18	2
mount	2.11r	10
mouseconfig	4.26	1
mozilla-nspr	1.0.1	26
mozilla-nss	1.0.1	26
mpage	02/05/02	4
mtools	03/09/08	5
mtr	0.49	7
mt-st	0.7	6
mtx	01/02/16	5

## Anexos - Software de Referencia para el Servidor

NOMBRE	VERSIÓN	REVISIÓN
namazu	2.0.10	8
namazu-cgi	2.0.10	8
nc	1.1	16
ncftp	03/01/03	6
ncompress	04/02/04	31
ncpfs	2.2.0.18	11
ncurses4	5	9
ncurses	5.2	28
netconfig	0.8.12	3
netpbm	9.24	9.80.2
net-snmp	5.0.6	8.80.2
net-snmp-utils	5.0.6	8.80.2
net-tools	1.6	7
newt	0.51.0	1
nfs-utils	1.0.1	2
nhp	1.42	3
nkf	1.92	11
nmap	3	1
nscd	02/03/02	4.80.6
nss_ldap	198	3
ntp	4.1.1a	9
ntsysv	01/03/06	3
nut-cgi	0.45.4	5
open	1.4	16
openjade	01/03/01	9
openldap12	01/02/13	9
openldap	2.0.27	02/08/00
openldap-clients	2.0.27	02/08/00
openldap-servers	2.0.27	02/08/00
openssh	3.4p1	2
openssh-clients	3.4p1	2
openssh-server	3.4p1	2
opensslo95a	0.9.5a	21
opensslo96	0.9.6	16.8
openssl	0.9.6b	33
openssl-perl	0.9.6b	33
ORBit	0.5.13	5
ORBit2	02/04/01	1
orbit-python	1.99.0	4
pam	0.75	46.8.0
pam_krb5	1.56	1
pam_smb	01/01/06	5
pango	01/01/01	1
parted	01/04/24	6
passwd	0.67	3
patch	02/05/04	14
pciutils	02/01/10	2
pcre	3.9	5
pdksh	05/02/14	19

## Anexos - Software de Referencia para el Servidor

NOMBRE	VERSIÓN	REVISIÓN
perl	05/08/00	55
perl-Crypt-SSLeay	0.45	2
perl-DateManip	5.4	27
perl-Digest-SHA1	2.01	6
perl-File-MMAGIC	1.15	2
perl-Filter	1.28	9
perl-Frontier-RPC	0.06	33
perl-HTML-Parser	3.26	14
perl-HTML-Tagset	3.03	25
perl-libwww-perl	5.65	2
perl-libxml-eno	1.02	25
perl-libxml-perl	0.07	25
perl-Mail-SpamAssassin	2.53	1
perl-NKF	1.71	7
perl-Parse-Yapp	1.05	26
perl-Text-Kakasi	1.05	2
perl-TimeDate	1.13	2
perl-Time-HiRes	1.2	23
perl-URI	1.21	3
perl-XML-Dumper	0.4	22
perl-XML-Encoding	1.01	20
perl-XML-Grove	0.46alpha	21
perl-XML-Parser	2.31	12
pinfo	0.6.4	7
pnm2ppa	1.04	5
popt	01/07/01	1.8x
portmap	4	46
ppp	02/04/01	7
prelink	0.2.0	8
procinfo	18	5
procmail	3.22	7
procps	2.0.7	25
psmisc	20.2	6
pspell	0.12.2	14
psutils	1.17	17
pwlib	01/03/03	5
pygtk2	1.99.12	7
pygtk2-libglade	1.99.12	7
pyOpenSSL	0.5.0.91	1
python	02/02/01	17
python-optik	1.3	2
pyxf86config	0.3.1	2
quota	3.06	5
raidtools	1.00.2	3.3
rc	01/04/01	0.ximian.5.1
rcd	01/04/03	0.ximian.5.1
rdate	1.2	5
rdist	06/01/05	24
readline41	4.1	14

## Anexos - Software de Referencia para el Servidor

NOMBRE	VERSIÓN	REVISIÓN
readline	4.3	3
recode	3.6	6
red-carpet	2.0.1	0.ximian.5.1.1
redhat-config-packages	1.0.1	1
redhat-logos	01/01/06	2
redhat-menus	0.26	1
redhat-release	8	8
redhat-switchmail	0.5.14	1
redhat-switch-printer	0.5.12	1
reiserfs-utils	03/06/02	2
rhmask	1.2	2
rhn-applet	2.0.9	0.8.0.1
rhlib	1	1
rhpl	0.51	1
rootfiles	7.2	4
rpm404-python	4.0.4	8x.27
rpm	04/01/01	1.8x
rpm-python	04/01/01	1.8x
rp-pppoe	3.4	7
rsh	0.17	10
ruby-docs	01/06/07	10
samba	02/02/07	05/08/00
samba-client	02/02/07	05/08/00
samba-common	02/02/07	05/08/00
sane-backends	1.0.8	5
sash	3.4	14
screen	03/09/11	10
scrollkeeper	0.3.10	7
sed	3.02	13
setserial	2.17	9
setup	02/05/20	1
setuptools	1.1	1
sgml-common	0.6.3	12
shadow-utils	20000902	12.8
shapecfg	02/02/12	10
sh-utils	2.0.12	3
slang	01/04/05	11
slocate	2.6	4
spamassassin	2.53	1
spamassassin-tools	2.53	1
stat	3.3	4
statserial	1.1	30
sudo	01/06/06	1
symlinks	1.2	16
sysklogd	01/04/01	10
syslinux	1.75	3
sysstat	4.0.5	3
SysVinit	2.84	5
talk	0.17	17

## Anexos - Software de Referencia para el Servidor

NOMBRE	VERSIÓN	REVISIÓN
tar	1.13.25	8
tcp_wrappers	7.6	23
tcsch	6.12	2
telnet	0.17	23
termcap	11.0.1	13
textutils	2.0.21	5
tftp	0.29	3
time	1.7	19
timeconfig	03/02/09	1
tmpwatch	02/08/04	3
traceroute	1.4a12	6
tree	1.2	20
ttcp	1.12	5
ttfprint	0.9	6
unix2dos	2.2	17
up2date	3.0.7	1
up2date-gnome	3.0.7	1
usbutils	0.9	7
usermode	1.63	1
usermode-gtk	1.63	1
utempter	0.5.2	10
util-linux	2.11r	10
vim-common	6.1	18.8x.1
vim-enhanced	6.1	18.8x.1
vim-minimal	6.1	18.8x.1
vixie-cron	3.0.1	69
vlock	1.3	11
vnc-doc	3.3.3r2	39.2
w3c-libwww	05/04/00	1
w3c-libwww-apps	05/04/00	1
w3m-el-common	01/03/01	1
watanabe-vf	1	8
webalizer	2.01_10	9
wget	01/08/02	5
which	2.14	1
whois	1.0.10	4
whoson	2.02a	1
wireless-tools	25	1
wl-common	02/08/01	8
Wnn6-SDK	1	21
words	2	20
wvdial	1.53	7
xferstats	2.16	3
XFree86-base-fonts	04/02/00	72
XFree86-doc	04/02/00	72
XFree86-font-utils	04/02/00	72
XFree86-libs	04/02/00	72
XFree86-Mesa-libGL	04/02/00	72
XFree86-Mesa-libGLU	04/02/00	72

## Anexos - Software de Referencia para el Servidor

<b>NOMBRE</b>	<b>VERSIÓN</b>	<b>REVISIÓN</b>
XFree86-xfs	04/02/00	72
Xft	2	1
xinetd	02/03/11	01/08/00
xml-common	0.6.3	12
ypbind	1.11	2
yp-tools	2.7	3
zisosfs-tools	1.0.3	5
zlib	01/01/04	8.8x
zsh	4.0.4	8

# Anexos - Código para la Instalación del Escritorio

## Anexo F. Código para la instalación del escritorio

Este código se usa para instalar un escritorio con el francés como idioma predeterminado. Usa el programa *kickstart* de Red Hat para la instalación y *Red Carpet* de Ximian para las actualizaciones. Asimismo, está adaptado para una instalación de un hardware y una configuración de máquina determinadas, incluyendo una tarjeta de video específica y la configuración de red; habría que hacer algunos cambios para atender a la instalación concreta de la Administración. El código que se muestra aquí no es más que un ejemplo para indicar que la funcionalidad se puede construir, y aunque debería funcionar, no cuenta con garantía alguna por parte de **netproject**.

```
lang fr_FR
langsupport fr_FR

# Other languages may need other keyboards
keyboard uk

# This defines the type of mouse attached to the client.
mouse --emulthree genericps/2

# This needs changing to the location of the desktop.
timezone --utc Europe/London

# The password shown here would need changing
rootpw --iscrypted $1$7QNhVztt$2/DrxHONbGs91.D5k4rx21
reboot
install

# Change 192.168.1.1 to the IP address of the server.
# Change /mnt/space/RedHat-8.0 to the location on the server of the
# software repository.
nfs --server 192.168.1.1 --dir /mnt/space/RedHat-8.0
bootloader --location=mbr --append vga=0x305
zerombr yes
clearpart --linux -initlabel
part / --fstype ext3 --size 256 --grow --maxsize 512
part /usr --fstype ext3 --size 927 --grow --maxsize 2048
part swap -recommended
network --bootproto dhcp

# Change 192.168.1.1 to the servers IP address
auth --useshadow --enablemd5 --enableldap --enableldapauth --ldapserver
192.168.1.1 --ldapbasedn dc=netproject,dc=com
firewall --disabled

# This might need changing to reflect the monitor being used.
xconfig --depth 16 --resolution 1024x768 --defaultdesktop=GNOME --startxonboot
--card "S3 Savage (generic)" --videoram 16384

%packages -resolvedeps
librpm404
redhat-artwork
gnome-session
XFree86
gdm
openoffice-libs
openoffice-il8n
libglade
gdk-pixbuf
Gconf
compat-libstdc++
indexhtml
libcapplet0
gdk-pixbuf-gnome
libghttp
mozilla-nss
```



## Anexos - Código para la Instalación del Escritorio

```
metacity
nautilus
gnome-panel
control-center
XFree86-75dpi-font
gftp
gedit
emacs
desktop-backgrounds-extra



```

%pre --interpreter /usr/bin/python
import os, fcntl, CDROM
def eject(file):
    try:
        f = os.open(file, os.O_RDONLY | os.O_NONBLOCK)
    except OSError, (errno, strerror):
        print "%s - OS error(%s): %s" % (file, errno, strerror)
        return
    while 1:
        try:
            fcntl.ioctl(f, CDROM.CDROMEJECT);
        except IOError, (errno, strerror):
            print "%s - I/O error(%s): %s" % (file, errno, strerror)
            if (errno == 16):
                continue
            break
    os.close(f)
    eject("/dev/hda")
    eject("/dev/hdb")
    eject("/dev/hdc")
    eject("/dev/hdd")

%post
/bin/sh << "EOF" >> /root/preboot.log 2>&1
MNT=/mnt/tmp
# Change 192.168.1.1 to the servers IP address
HOST=192.168.1.1
date
mkdir -p $MNT
mount $HOST:/kickstart $MNT -t nfs -o ro
cd $MNT
./postinst.sh preboot
EOF

```



The code above calls a script called postinst.sh which contains the following:



```
#!/bin/sh
# description: Installation & upgrade script
# chkconfig: 2345 25 03
```


```

## Anexos - Código para la Instalación del Escritorio

```
# Â© netproject 2003
# Sean Atkinson <sean@netproject.com>, March 2003
SERVICE=postinst
case $1 in
start)
exec $0 postboot >> /root/postboot.log 2>&1
;;
stop)
exit 0
;;
preboot)
rpm -U *.rpm
mkdir /kickstart/
cp -v diffs.patch gdm_bg.png gdm_logo.png /kickstart/
cp -v grub_splash.xpm.gz /boot/grub/
cp -v postinst.sh /etc/rc.d/init.d/$SERVICE
chkconfig --add $SERVICE
exit 0
;;
postboot)
;;
*)
echo "Usage: $0 start|stop|preboot|postboot"
exit 1
;;
esac
# Change 192,168.1.1 to servers IP address. Also details need to reflect
# the actual location and version of the distribution used
HOST=192.168.1.1
RCD=http://$HOST/rcd/
REL=redhat-80-i386
date
# The rpm commands below refer to specific versions of packages. the
# latest versions should be substituted if needed.
rpm -U $RCD/redcarpet/$REL/rcd-1.2.1-1.ximian.3.i386.rpm
cat << EOF >> /etc/ximian/rcd.conf
[Network]
host=$RCD
[Cache]
enabled=false
EOF
service rcd start
rpm -U $RCD/redcarpet/$REL/rc-1.2.1-1.ximian.1.i386.rpm
```

## Anexos - Código para la Instalación del Escritorio

```
until rc ping
do
sleep 1
done
rc subscribe $REL redcarpet ximian-evolution
service rcd restart
rpm -e kernel-pcmcia-cs
rpm -e kudzu
rpm -e comps
rpm -e authconfig
until rc ping
do
sleep 1
done
rc update --no-confirmation
for PKG in galeon evolution xpdf vnc vnc-doc openoffice
do
rc install --no-confirmation $PKG
done
# This command assumes a specific release of the java run time
# environemnt
ln -s /usr/java/j2re1.4.1_02/plugin/i386/ns610/libjavaplugin_oji.so \
/usr/lib/mozilla-1.0.1/plugins/
patch -p0 < /kickstart/diffs.patch
chkconfig --del $SERVICE
chkconfig rcd off
for DIR in /tmp /var/tmp
do
rm -rf $DIR
ln -s /dev/shm $DIR
done
reboot

The code above uses the contents of a file called diffs.patch, which contains:
--- gdm.conf 2003-03-25 14:01:20.000000000 +0000
+++ /etc/X11/gdm/gdm.conf 2003-03-25 14:02:38.000000000 +0000
@@ -21,7 +21,7 @@
  Chooser=/usr/bin/gdmchooser
  DefaultPath=/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin
  DisplayInitDir=/etc/X11/gdm/Init
-Greeter=/usr/bin/gdmgreeter
+Greeter=/usr/bin/gdmlogin
  #Uncomment this for the regular greeter
  #Greeter=/usr/bin/gdmlogin --disable-sound --disable-crash-dialog
```

## Anexos - Código para la Instalación del Escritorio

```
RemoteGreeter=/usr/bin/gdmlogin
@@ -99,7 +99,7 @@
GlobalFaceDir=/usr/share/faces/
Icon=/usr/share/pixmaps/gdm.xpm
LocaleFile=/etc/X11/gdm/locale.alias
-Logo=
# This assumes the png file exists in the location shown. Change as
# required.
+Logo=/kickstart/gdm_logo.png
## nice RH logo for the above line: /usr/share/pixmaps/redhat/shadowman-200.png
Quiver=true
SystemMenu=true
@@ -114,8 +114,8 @@
PositionY=0
XineramaScreen=0
#Type can be 0=None, 1=Image, 2=Color
-BackgroundType=0
-BackgroundImage=
+BackgroundType=1
# This assumes the png file exists in the location shown. Change as-
# required.
+BackgroundImage=/kickstart/gdm_bg.png
BackgroundScaleToFit=true
BackgroundColor=#27408b
BackgroundRemoteOnlyColor=true
--- grub.conf 2003-03-25 14:01:20.000000000 +0000
+++ /boot/grub/grub.conf 2003-03-25 14:01:20.000000000 +0000
@@ -9,7 +9,7 @@
#boot=/dev/hda
default=0
timeout=10
-splashimage=(hd0,0)/boot/grub/splash.xpm.gz
# This assumes the xpm.gz file exists in the location shown. Change as
# required.
+splashimage=(hd0,0)/boot/grub/grub_splash.xpm.gz
# This assumes a particular version of the kernel ie 2.4.18-27.8.0.
# Change all references as needed.
title Red Hat Linux (2.4.18-27.8.0)
    root (hd0,0)
    kernel /boot/vmlinuz-2.4.18-27.8.0 ro root=LABEL=vga=0x305
--- fstab 2003-03-27 10:52:12.000000000 +0000
+++ /etc/fstab 2003-03-27 10:54:01.000000000 +0000
@@ -1,8 +1,9 @@
```

## Anexos - Código para la Instalación del Escritorio

```
LABEL=/          /          ext3          defaults          1 1
none            /dev/pts    devpts        gid=5,mode=620    0 0
none            /proc       proc          defaults          0 0
-none           /dev/shm    tmpfs        defaults          0 0
-LABEL=/usr     /usr        ext3          defaults          1 2
+none           /dev/shm    tmpfs        defaults,noexec   0 0
+LABEL=/usr     /usr        ext3          defaults,ro       1 2
/dev/hda3       swap        swap          defaults          0 0
/dev/cdrom      /mnt/cdrom  iso9660       noauto,owner,kudzu,ro 0 0
/dev/fd0        /mnt/floppy auto          noauto,owner,kudzu 0 0
# Change 192.168.1.1 to the IP address of the NFS server.
+192.168.1.1:/hom /home      nfs          defaults,noexec   0 0
e
```

## Anexos - Glosario

### Anexo G. Glosario

<b>ACL</b>	Lista de Control de Acceso ( <i>Access Control List</i> ). Una Lista de Control de Acceso es una lista adjunta a un objeto como, por ejemplo, un fichero. Consiste en expresiones de control, cada una de las cuales concede o niega alguna capacidad a un usuario particular o un grupo de usuarios.
<b>Administración</b>	Una administración pública europea.
<b>Administradores</b>	La gestión de TI de una administración.
<b>Aleatorizaciones</b>	V. Hashes
<b>API</b>	Interfaz de Programación de Aplicaciones ( <i>Application Program Interface</i> ). El método concreto prescrito por un sistema operativo informático, aplicación o herramienta de terceros por la cual un programador que escribe un programa de aplicación puede hacer llamadas al sistema operativo. También conocido como Interfaz de Programadores de Aplicaciones.
<b>Applet Java</b>	Un mini-programa de software que un navegador Java o ActiveX habilitado descarga y usa automáticamente. Puede añadir un soporte sofisticado a las páginas Web, mucho más allá de la programación como DHTML o Javascript.
<b>ASP</b>	<i>Active Server Pages</i> . Una página HTML que incluye uno o más scripts (pequeños programas intercalados) que se procesan en un servidor Web de Microsoft antes de enviar la página al usuario. Una ASP es algo similar a una aplicación de servidor o a una aplicación de interfaz común de pasarela (CGI) que implica que los programas se ejecutan en el servidor, normalmente adaptando a medida una página para el usuario.
<b>BDC</b>	Controlador de dominios de reserva ( <i>Backup Domain Controller</i> ). Papel que puede asignarse a un servidor de una red de ordenadores que usan el sistema operativo Windows NT. Windows NT utiliza el concepto de dominio para gestionar el acceso a un conjunto de recursos en red (aplicaciones, impresoras, y otros recursos) para un grupo de usuarios. El usuario sólo necesita conectarse al dominio para obtener acceso a los recursos, que pueden encontrarse en distintos servidores en la red. Un servidor, conocido como controlador principal de dominio (PDC), gestiona la base de datos maestra de usuarios del dominio. Otro u otros servidores se conocen como controladores de dominios de reserva. Periódicamente, el controlador principal de dominios envía copias de la base de datos a los controladores de dominios de reserva. Un controlador de dominios de reserva puede ascender a controlador principal de dominios si el servidor PDC falla, y también puede ayudar a equilibrar la carga de trabajo si la red está bastante saturada.

## Anexos - Glosario

<b>Binario</b>	El software suele redactarse en un lenguaje fácilmente comprensible por el hombre que se denomina Código Fuente. Este código se convierte en una forma que el procesador del ordenador entiende directamente, y recibe el nombre de Binario porque consiste en una serie de ceros y unos. Es la forma en que se entrega la mayor parte del código propietario y resulta muy difícil volverlo a convertir en una forma fácilmente comprensible por el hombre. Disponer del código fuente permite hacer cambios en el software y que la gente entienda lo que está haciendo.
<b>CIL</b>	Lenguaje Intermedio Común ( <i>Common Intermediate Language</i> ). Código intermedio independiente de máquina y compilador que funciona mediante un <i>runtime</i> de Lenguaje Común (CLR). Este código se puede sacar de ciertos lenguajes compilados, como el C# y el C. Tanto el CIL como el CLR forman parte de la Infraestructura de Lenguajes Comunes (CLI)
<b>Código Beta</b>	Cuando el software está redactado pasa a través de varias etapas diferentes antes de que se lo considere lo bastante libre de errores y funcionalmente correcto para ser usado como software de producción. La primera de estas etapas se denomina alpha y la segunda beta. Por lo tanto el Código Beta es el sustancialmente correcto pero aún podría tener errores significativos. Así pues, habría de usarse con precaución
<b>Código fuente Daemon</b>	V. Binario Un programa o proceso relativo a un sistema que permanece en situación de baja prioridad, en segundo plano, hasta que es llamado por otro proceso o algún evento para desempeñar su tarea.
<b>DBMS</b>	Sistema de Gestión de Bases de Datos ( <i>Data Base Management System</i> ). Un programa que permite a uno o varios usuarios de ordenador crear datos y acceder a ellos en una base de datos. El DBMS gestiona las peticiones del usuario (y las peticiones procedentes de otros programas) para que los usuarios y otros programas no tengan que entender dónde están físicamente los datos en los soportes de almacenaje y, en un sistema multiusuario, quién más puede estar accediendo a los datos.
<b>DHCP</b>	Protocolo de Configuración Dinámica del Host ( <i>Dynamic Host Configuration Protocol</i> ). Protocolo de comunicaciones que permite que los administradores de la red gestionen de manera central o automática la asignación de las direcciones de Protocolo de Internet (PI) en la red de una empresa u organización
<b>Disparador</b>	( <i>Trigger</i> ) Un conjunto de instrucciones del Lenguaje Estructurado de Consulta (SQL) que “dispara” automáticamente una acción cuando se da una operación concreta, como por ejemplo el cambio datos en una tabla.

## Anexos - Glosario

<b>Distribución</b>	En el caso del software de fuentes abiertas, Linux, por ejemplo, empresas como Red Hat se especializan en empaquetar componentes a partir de diversas fuentes en un paquete o conjunto de paquetes que se pueden distribuir a los usuarios como una descarga única o como un conjunto de CDs.
<b>DNS</b>	Servidor de Nombres Dominios ( <i>Domain Name Server</i> ). Se utiliza para la conversión entre el nombre de la máquina en Internet y su dirección numérica.
<b>Dominio (Autenticación)</b>	Un conjunto de identificadores de autorización (personas y procesos) gestionado por un servidor de autenticaciones.
<b>Entradas “boilerplate”</b>	Cualquier parte de datos corrientes y no diferenciados, normalmente “relleno” que tiene que estar ahí pero es de escaso interés.
<b>Equilibrado de carga</b>	El equilibrado de carga consiste en dividir la cantidad de trabajo que un ordenador tiene que hacer entre dos o más procesadores u ordenadores para hacer más trabajo en el mismo tiempo y, en general, servir a los usuarios con mayor rapidez. El equilibrado de carga se puede aplicar al hardware, al software o a una combinación de los dos. Lo normal es que el equilibrado de carga sea la principal razón del agrupamiento de los servidores de ordenadores.
<b>Escenario</b>	Guión. Argumento. Ver capítulo 4.
<b>Esquema</b>	La organización o estructura de una base de datos. La actividad del modelado de datos desemboca en un esquema.
<b>FTP</b>	Protocolo de Transferencia de Archivos ( <i>File Transfer Protocol</i> ). Un método independiente del sistema que sirve para transmitir (enviar y recibir) archivos entre sistemas conectados vía TCP/IP. Asegura que el archivo se ha transferido correctamente, incluso si hay errores durante la transmisión.
<b>Gestor de sesiones</b>	Cuando un usuario se conecta a un ordenador se crea una sesión que consiste en un entorno de control de la información, de toda una serie de procesos. El gestor permite al usuario que cambie este entorno y pueda también guardarlo para que cuando vuelva a entrar en el ordenador éste vuelva al estado que tenía antes de su última desconexión.
<b>Gestor de ventanas</b>	En un moderno entorno gráfico, se le presenta al usuario una serie de ventanas en las que se están desarrollando diversos procesos. Esto significa que pueden ejecutar muchas cosas distintas al mismo tiempo y mostrar los resultados simultáneamente en una pantalla. El papel del gestor de ventanas consiste precisamente en gestionar estas ventanas. Tiene que mantener el rastro de la ventana en la que el usuario está interesado en cada momento, permitirle al usuario que cambie de ventana y crear o eliminar ventanas. También controla qué aspecto tiene la ventana, su forma y las características de control.
<b>GPL</b>	Licencia Pública General GNU ( <i>GNU General Public License</i> ).
<b>GUI</b>	Interfaz Gráfica de Usuario.
<b>Hashes</b>	Un Hash es un identificador único resumido, la “huella digital” de algo más complicado. Los Hashes se crean usando funciones matemáticas unidireccionales. Se utilizan en los sistemas de bases de datos y en sistemas criptográficos y de seguridad.



## Anexos - Glosario

<b>HTTP</b>	Protocolo de Transferencia de Hipertexto ( <i>Hypertext Transfer Protocol</i> ). Un conjunto de normas para intercambiar archivos (de texto, gráficos, de vídeo, de audio y otros recursos multimedia) en el <i>World Wide Web</i> . Relacionado con la serie de protocolos TCP/IP (que son la base para el intercambio de información en Internet), HTTP es un protocolo de aplicación.
<b>JDBC</b>	Conectividad de Bases de Datos Java ( <i>Java Database Connectivity</i> ). Una especificación de interfaz de programación de aplicaciones (API) para conectar programas escritos en Java a los datos de las bases de datos más conocidas. La interfaz de programación de aplicaciones permite codificar las instrucciones de consultas de acceso hechas en Lenguaje Estructurado de Consulta (SQL) que entonces se pasan al programa que gestiona la base de datos. Devuelve los resultados a través de una interfaz similar.
<b>Kernel</b>	El núcleo o parte esencial de un sistema operativo que se ocupa de tareas como la asignación de memoria, la entrada y salida a los dispositivos, la asignación de los procesos, la seguridad y el acceso del usuario.
<b>LDAP</b>	Protocolo Ligero de Acceso a Directorio ( <i>Lightweight Directory Access Protocol</i> ). Un protocolo de software que permite la localización de entidades, particulares y otros recursos como archivos y dispositivos en una red, tanto en la Internet pública como en una Intranet interna. LDAP es una versión “ligera” (con menos código) del Protocolo de Acceso a Directorio (DAP), que forma parte de X.500, un estándar para servicios de directorio en una red.
<b>LGPL</b>	<i>GNU Lesser General Public License</i> .
<b>Licencia de usuarios potenciales</b>	Tipo de licencia que se cobra teniendo en cuenta el número máximo de usuarios que tienen la capacidad de acceder a una aplicación.
<b>Licencia de usuarios simultáneos</b>	Tipo de licencia que se cobra teniendo en cuenta el número máximo de usuarios que pueden acceder a una aplicación al mismo tiempo.
<b>MAA</b>	Agente de Acceso al Correo ( <i>Mail Access Agent</i> ). Término utilizado en este documento para describir al componente de correo de servidor que gestiona el acceso al buzón de correo por parte de un MUA. Se pueden citar como ejemplos los servidores POP e IMPA. En el Anexo C se trata más a fondo este tema.
<b>MDA</b>	Agente de Entrega del Correo ( <i>Mail Delivery Agent</i> ). Un componente de correo que acepta correo de un MTA y lo manda al buzón de correo.
<b>Metadatos</b>	Una definición o descripción de los datos.
<b>MTA</b>	Agente de Transferencia de Correo ( <i>Mail Transport Agent</i> ). Se trata del componente de correo responsable de decidir si el correo que maneja es para una cuenta local o no. Pasa el correo local a un MDA o lo almacena directamente en el propio buzón. El correo remoto se envía a otro MTA.
<b>MUA</b>	Agente de Correo de Usuario ( <i>Mail User Agent</i> ). El componente de correo del cliente que recupera el correo del buzón y lo presenta al usuario. Permite al usuario crear nuevo correo y enviarlo a un MTA para su transmisión. El MUA suele ir asociado a una interfaz gráfica.

## Anexos - Glosario

<b>.NET</b>	El conjunto de tecnologías de software que tiene Microsoft para conectar información, personas, sistemas y dispositivos. Se basa en los servicios web que son pequeñas aplicaciones que pueden conectarse entre sí y también a otras aplicaciones mayores en Internet. El proyecto MONO de OSS es una implementación del marco de desarrollo .NET
<b>NFS</b>	Servicio de Archivos en Red ( <i>Network File Service</i> ). Un protocolo de uso corriente en los sistemas tipo Unix para acceder a los archivos que se guardan en sistemas remotos como si fueran locales.
<b>ODBC</b>	Conectividad Abierta de Bases de Datos (Open Database Connectivity). Una especificación abierta de interfaz de programación de aplicaciones para acceder a una base de datos. Mediante el uso de las instrucciones de ODBC en un programa se puede acceder a los archivos de una serie de bases de datos diferentes como, entre otras, Access, dBase, DB2, Excel y Text. Además del software de ODBC, se necesita un módulo o controlador independiente para acceder a cada base de datos.
<b>OSS</b>	Véase Software de Fuentes Abiertas.
<b>Pantalla Verde</b>	Un terminal o visor que sólo puede visualizar caracteres de tamaño fijo y (posiblemente) gráficos de bloque sencillos. Su nombre se debe al hecho de que muchas pantallas de visualización de grandes equipos <i>mainframe</i> de las décadas de los 70 y los 80 usaban fósforo verde.
<b>PDA</b>	Asistente Personal Digital ( <i>Personal Digital Assistant</i> ). Ordenador manual.
<b>PDC</b>	Controlador Principal de Dominios ( <i>Primary Domain Controller</i> ). Véase BDC ( <i>Backup Domain Controller</i> - Controlador de dominios de reserva).
<b>PHP</b>	PHP: <i>Hypertext Preprocessor</i> (Preprocesador de Hipertextos). Lenguaje e intérprete en código máquina que está disponible de manera gratuita y que se puede usar principalmente en los servidores de Web Linux. El PHP constituye una alternativa a la tecnología de ASP de Microsoft. Como en el caso de la ASP, el código PHP se intercala en una página Web junto con su HTML. Antes de enviar la página a un usuario que la ha pedido, el servidor Web pide al PHP que interprete y realice la operación solicitada en el código PHP.
<b>PKI</b>	Infraestructura de Clave Pública (Public Key Infrastructure). Una PKI permite a usuarios de una red pública básicamente no segura, como Internet, intercambiar datos y dinero de forma segura y privada usando un par de claves criptográficas pública y privada que se obtienen y comparten a través de una entidad fiable. La infraestructura de clave pública facilita un certificado digital que puede identificar a un individuo o una entidad así como servicios de directorio que pueden almacenar y, en caso necesario, revocar los certificados.
<b>Procedimiento guardado</b>	Una serie de instrucciones de Lenguaje Estructurado de Consulta (SQL) con un nombre asignado que se almacena en la base de datos de forma compilada para que lo puedan compartir diferentes programas.

## Anexos - Glosario

<b>Protocolo</b>	Conjunto de reglas que utilizan los puntos extremos de una conexión de telecomunicaciones cuando están mutuamente conectados. Existen protocolos a distintos niveles en toda conexión de telecomunicaciones. Hay protocolos hardware de telefonía. También hay protocolos entre cada una de las distintas capas y cada capa correspondiente en el otro extremo de una comunicación. Los dos puntos extremos deben reconocer y respetar un protocolo. Los protocolos se suelen describir en una norma internacional o de la industria.
<b>Protocolo DEC</b>	Digital Equipment Corporation o DEC creó un conjunto de protocolos para controlar los dispositivos de terminales. Estos protocolos alcanzaron un amplio uso y ahora son estándares
<b>Relé abierto</b>	Un relé abierto (a veces llamado relé no seguro o relé a terceros) es un servidor de correo SMTP que permite el relé a terceros de mensajes de correo electrónico. Al procesar correo que no es para un usuario local ni procede de él, un relé abierto hace posible que un remitente poco escrupuloso distribuya grandes volúmenes de spam. De hecho, el propietario del servidor - que no suele ser consciente del problema - facilita recursos informáticos y de red que sirven al propósito del remitente. Además del coste económico que conlleva el que un spammer ataque a un servidor, una entidad también puede sufrir caídas del sistema, daños al equipo y pérdida de negocios.
<b>Servicios Gopher</b>	Antiguo sistema de recuperación de información de tipo hipertexto.
<b>Servidor proxy</b>	Servidor apoderado. Un servidor que actúa como intermediario entre un usuario de una estación de trabajo e Internet para que la organización garantice la seguridad, el control administrativo y el servicio de caché. Un servidor <i>proxy</i> se asocia en todo o en parte con un servidor de pasarela que separa la red de la organización de la red externa y un servidor de “cortafuegos” que protege a la red de la organización de toda intrusión externa.
<b>Servlet Java</b>	Programa Java que opera como parte de un servicio en red, normalmente un servidor HTTP, y responde a consultas de clientes. El uso más corriente de un <i>servlet</i> es ampliar un servidor web generando contenidos de web de forma dinámica. Por ejemplo, un cliente puede necesitar información de una base de datos; un <i>servlet</i> puede recibir por escrito la consulta, y obtiene y procesa los datos que necesita el cliente y luego envía el resultado al cliente.
<b>Sesión X</b>	Cuando un usuario entra en un ordenador y ejecuta programas mediante el protocolo X, se crea una sesión X.
<b>SMB</b>	Bloque de Mensajes del Servidor ( <i>Server Message Block</i> ). Es el protocolo usado en la red Windows para que los recursos de una máquina como, por ejemplo, los archivos, puedan ser compartidos por otras máquinas como si fueran locales.
<b>SMS</b>	Servicio de Mensajes Cortos ( <i>Short Message Service</i> ). Servicio para enviar mensajes de hasta 160 caracteres (224 caracteres si se usa un modo de 5 bits) a teléfonos móviles que se sirve de la norma <i>Global System for Mobile Communications</i> (GSM, Sistema Global para Comunicaciones Móviles).
<b>Software de fuentes abiertas</b>	Se define en: <a href="http://www.gnu.org/philosophy/free-sw.html">http://www.gnu.org/philosophy/free-sw.html</a>

## Anexos - Glosario

<b>SQL</b>	Lenguaje estructurado de consulta ( <i>Structured Query Language</i> ). Lenguaje interactivo que se utiliza para recuperar y actualizar la información contenida en una base de datos. Aunque el SQL es un estándar tanto ANSI como ISO, muchos productos de bases de datos respaldan el SQL con ampliaciones propietarias del lenguaje estándar. Las consultas toman la forma de un lenguaje de comando que permite al usuario seleccionar, insertar, actualizar, encontrar la ubicación de los datos, etc. También es una interfaz de programación.
<b>SSL</b>	( <i>Capa de conexión segura</i> ) <i>Secure Sockets Layer</i> . Protocolo de uso muy extendido para gestionar la seguridad de una transmisión de mensajes en Internet. Recientemente a SSL le ha sucedido con éxito el protocolo TLS, que se basa en SSL. SSL usa una capa de programa situada entre las capas del Protocolo de Transferencia de Hipertexto (HTTP) y el Protocolo de Control de Transporte (TCP). SSL se incluye como parte de los navegadores Microsoft y Netscape y la mayoría de los productos para servidores Web.
<b>Tarjeta inteligente</b>	( <i>Smart Card</i> ) Una tarjeta de plástico que contiene un chip informático. La tarjeta se usa para realizar operaciones que requieren los datos almacenados en el chip.
<b>Terminal X</b>	Un terminal diseñado especialmente para ejecutar un servidor X que permite a los usuarios visualizar los resultados de los programas que están operando en otro ordenador utilizando el protocolo X en una red.
<b>TLS</b>	Seguridad de Capa de Transporte ( <i>Transport Layer Security</i> ). Una capa que proporciona servicios de cifrado y autenticación que se pueden negociar durante la fase de puesta en marcha de muchos protocolos de Internet (por ejemplo SMTP, LDAP, IMAP, POP3). TLS deriva de SSL y utiliza los mismos certificados, aunque no exige que cada servicio que se vaya a prestar lleve un nuevo número de puerto; véase SSL.
<b>VMS</b>	Sistema operativo creado por Digital Equipment Corporation (DEC) para ser usado en sus mini-ordenadores VAX. Posteriormente fue portado al sistema Alpha 64-bit. Uno de los principales creadores de VMS diseñó posteriormente el núcleo de Windows NT.
<b>WebDAV</b>	Versiones y Autorías Distribuidas en la WEB ( <i>World Wide Web Distributed Authoring and Versioning</i> ). Norma del Equipo de Trabajo de Ingeniería de Internet (IETF) para una Autoría “colaboradora” en la Web: un conjunto de ampliaciones del protocolo de Transferencia de Hipertextos (HTTP) que facilita la edición “colaboradora” y la gestión de archivos entre usuarios situados a distancia uno de otros en Internet.
<b>XML</b>	Lenguaje extensible de marcado ( <i>Extensible Markup Language</i> ). Un modo flexible de crear formatos de información flexibles y de compartir tanto el formato como los datos en el <i>World Wide Web</i> , en las intranets y en cualquier otro lugar. XML es una recomendación formal del <i>World Wide Web Consortium</i> (W3C) semejante al lenguaje de las páginas web actuales, el HTML (Lenguaje de marcado de hipertexto).