

# **GUÍA CUBANA DE MIGRACIÓN A SOFTWARE LIBRE**

**Versión 0.1**

## **Sobre la presente**

Se presenta la versión 0.1 del documento: GUÍA CUBANA PARA LA MIGRACIÓN A SOFTWARE LIBRE. Para su correcta implementación él mismo deberá ir acompañado de:

- Documental Software Libre para estudiantes.
- El documento oficial con los detalles técnicos sobre la migración de Servidores.
- El documento oficial con los detalles técnico sobre la migración de Bases de Datos.
- El documento de Arquitecturas de referencia.
- LiveCDs de instalación de Nova, Debian, Ubuntu.
- Repositorios de aplicaciones libres.
- Sistclon.
- OCS Inventory.
- Servi-Desk.

## Los Autores

Ing. Ramón Paumier Samón.
---------------------------

Ing. Yoandy Pérez Villazón.
-----------------------------

Ing. Abel Meneses Abad.
-------------------------

## Colaboradores.

Est. Amaury Pérez
-------------------

Est. Marcos Ortiz
-------------------

Ing. Michel Hernández
-----------------------

Ing. Dayron Pérez
-------------------

Ing. Yaima Oval
-----------------

## Historial de Versiones.

<b>Versión</b>	<b>Detalles</b>		<b>Autor</b>
<b>0.1</b>	<ul style="list-style-type: none"><li>● <b>Versión Original</b></li></ul>		<b>RPS</b>
<b>0.2</b>	<ul style="list-style-type: none"><li>● <b>Se agrega ventajas de Distro Debian.</b></li><li>● <b>Se incluye propuesta de estructura organizativa para realizar la migración.</b></li></ul>		<b>RPS</b>

# Indice

Sobre la presente.....	2
Los Autores.....	3
Historial de Versiones.....	4
1- UNA INTRODUCCIÓN NECESARIA.....	9
1.1- Las libertades.....	9
1.2- Las distribuciones.....	10
1.3- La gran rivalidad.....	11
1.4- Panorama Mundial.....	13
1.5- Cuba.....	16
1.6- Concluyendo esta parte.....	18
2- CATEGORÍAS, LICENCIAS Y ESTÁNDARES.....	21
2.1- Categorías de Software Libre.....	21
2.1.1- Software de código abierto (open source).....	21
2.1.2- Software de dominio público.....	22
2.1.3- Software protegido con copyleft.....	22
2.1.4- Copyleft y la GNU GPL.....	22
2.1.5- Software libre no protegido con copyleft.....	24
2.1.6- Software abarcado por GPL.....	24
2.1.7- Software GNU.....	25
2.1.8- Software semilibre.....	25
2.1.9- Software propietario.....	25
2.1.10- Freeware.....	25
2.1.11- Shareware.....	26
2.1.12- Software comercial.....	27
2.2- Licencias de Software Libre.....	28
2.2.1- Software bajo licencia BSD.....	28
2.2.2- Software bajo licencia Apache.....	29
2.2.3- Software bajo licencia Mozilla.....	29
2.2.4- Software bajo licencia GPL 3.....	30
2.3- Nociones de Estándares Informáticos.....	30
2.3.1- Categorización de estándares según apertura y exclusividad.....	31
2.3.2- Un “NO estándar”.....	32
2.3.3- Estándar cerrado.....	32
2.3.4- Estándar RAND.....	33
2.3.5- Estándar abierto.....	34
2.3.6- Estándar libre.....	35
2.3.7- Categorización de estándares según carácter legal.....	35
2.3.8- Estándar legal.....	36
2.3.9- Estándar nacional.....	36
2.3.10- Estándar internacional.....	36
2.3.11- Estándar industrial.....	36
3- PENSANDO EN EL FUTURO.....	38
3.1- Primeras transformaciones.....	38
3.2- Las reacciones.....	41
3.3- El proceso ideal.....	43

4- METODOLOGÍA 2008.....	49
4.1- Descripción de los flujos de trabajo.....	50
4.1.1- Flujo de trabajo: Evaluación.....	50
4.1.2- Flujo de trabajo: Diseño.....	51
4.1.3- Flujo de trabajo: Pilotos.....	53
4.1.4- Flujo de trabajo: Formación.....	54
4.1.5- Flujo de trabajo: Implementación.....	56
4.1.6- Flujo de trabajo: Asistencia y Soporte Técnico.....	57
4.2- Alternativas de migración.....	59
5- ETAPA DE PREPARACIÓN.....	62
5.1- Levantamiento de Información.....	63
5.2- Plan de migración.....	66
5.3- Plan de acción institucional.....	67
5.3.1 Estructura organizativa de una gran Entidad para la Migración.....	67
5.3.2 Estructura organizativa de una pequeña Entidad para la Migración.....	68
5.4- Laboratorio de formación, capacitación y soporte.....	68
6- ETAPA DE MIGRACIÓN PARCIAL.....	71
6.1- Migración Parcial 1 de Servidores y Bases de Datos.....	71
6.1.1- Migración Parcial de Servidores.....	73
6.1.1.1- Propuestas de migración de un servidor DNS .....	73
6.1.1.2- Propuestas de migración hacia un servidor ldap. ....	76
6.1.1.3- Propuesta para migrar el DHCP .....	78
6.1.1.4- Propuesta para servicios de impresión.....	79
6.1.1.5- Propuestas de migración de un servidor de correo.....	81
6.1.1.6- Propuestas de migración de un servidor Proxy.....	81
6.1.1.7- Propuestas de migración de un servidor web.....	82
6.1.1.8- Propuestas de una herramienta de trabajo en grupo.....	83
6.1.1.9- Asterisk, alternativa libre, servidor de VoIP.....	83
6.1.1.10- Métodos de seguridad en Servidores GNU/Linux.....	84
6.1.1.11- Herramientas de seguridad en Servidores GNU/Linux.....	86
6.1.2- Migración Parcial de Bases de Datos. ....	89
6.2- Migración Parcial 2 Instalación de aplicaciones libres en el entorno propietario.....	93
6.3- Migración Parcial 3 de Escritorios y Entornos de Trabajo.....	94
6.3.1- Softwares equivalentes.....	94
6.3.1.1- Oficina.....	94
6.3.1.2- Gráficos.....	96
6.3.1.3- Herramientas y utilidades.....	97
6.1.3.4- Desarrollo.....	100
6.1.3.5- Antivirus.....	101
6.1.3.6- Seguridad.....	102
6.1.3.7- Aplicaciones Críticas.....	105
6.3.2- Distribuciones a utilizar en Escritorios y Servidores.....	106
6.3.2.1- Nova: Distribución cubana de Software Libre.....	106
6.3.2.2- Debian GNU/Linux .....	108
6.3.2.3- Ubuntu.....	110
6.3.3- Clonación de la Imagen a varias computadoras: Sistclon.....	112
6.4- Corrección de Errores.....	114
6.5- Metapaquete de cursos de formación.....	114

7- ETAPA DE MIGRACIÓN TOTAL.....	118
7.1- Migración Total 1 de Servidores y Bases de Datos.....	118
7.2- Migración Total 2 Instalación de aplicaciones libres en el entorno propietario.....	118
7.3- Migración Total 3 de Escritorios y Entornos de Trabajo.....	118
8- ETAPA DE CONSOLIDACIÓN.....	120
8.1- Formación y Certificación de usuarios.....	120
8.2- Soporte y Mantenimiento: Servi-Desk.....	120
ANEXO 1 HERRAMIENTAS PRIVATIVAS Y SUS EQUIVALENTES LIBRES.....	124
ANEXO 2 .....	127
REFERENCIAS BIBLIOGRÁFICAS.....	129

# **Parte 1**

# **UNA INTRODUCCIÓN**

# **NECESARIA.**

# 1- UNA INTRODUCCIÓN NECESARIA.

Entre los años 60 y 70 del Siglo XX, el software no era considerado un producto sino un añadido, que los vendedores de los grandes computadores de la época (los *mainframes*) aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. Este comportamiento era particularmente habitual en algunos de los mayores grupos de usuarios de la época, como DECUS (grupo de usuarios de computadoras DEC). A finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia.

Con este antecedente, en 1984 Richard Stallman comenzó a trabajar en el proyecto GNU, y un año más tarde fundó la Free Software Foundation, en lo adelante FSF. Stallman introdujo una definición para *free software* y el concepto de "*copyleft*", el cual desarrolló para dar a los usuarios libertad y para restringir las posibilidades de apropiación del software. (1)

El Software Libre es aquel que puede ser distribuido, modificado, copiado y usado; por lo tanto, debe venir acompañado del código fuente para hacer efectivas las libertades que lo caracterizan. Es conveniente no confundir el Software Libre con el software gratuito, éste no cuesta nada, hecho que no lo convierte en Software Libre, porque no es una cuestión de precio, sino de libertad.

## 1.1- Las libertades.

Algunas personas utilizan los términos "libre" (*Libre software*) y "gratis" (*Gratis software*) para evitar la ambigüedad de la palabra inglesa "free". Sin embargo, estos términos alternativos son usados únicamente dentro del movimiento del Software Libre, aunque están extendiéndose lentamente hacia el resto del mundo. Otros defienden el uso del término *open source software* (software de código abierto, también llamado de fuentes abiertas).

El movimiento del Software Libre hace especial énfasis en los aspectos morales o éticos del software, viendo la excelencia técnica como un producto secundario deseable de su estándar ético. El movimiento Open Source ve la excelencia técnica como el objetivo prioritario, siendo la compartición del código fuente un medio para dicho fin. Por dicho motivo, la Free Software Foundation se distancia tanto de este movimiento.

Software Libre es cualquier programa cuyos usuarios gocen de estas libertades:

- Libertad 0: la libertad para ejecutar el programa sea cual sea nuestro propósito.
- Libertad 1: la libertad para estudiar el funcionamiento del programa y adaptarlo a las necesidades (el acceso al código fuente es condición indispensable para esto).
- Libertad 2: la libertad para redistribuir copias y ayudar al resto.
- Libertad 3: la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad (el acceso al código fuente es condición indispensable para esto). (2)

GNU/LINUX es un proyecto de más de 20 años en desarrollo, que se asienta sobre una

base de cientos de programadores de todas partes del mundo. Es a su vez, el primer sistema operativo basado en UNIX que es 100% Software Libre. Anteriormente había otros sistemas operativos de libre distribución, aunque estos no eran totalmente Software Libre, ya que eran regidos por licencias más restrictivas.

## 1.2- Las distribuciones.

La base de todo sistema de Software Libre es un núcleo monolítico llamado GNU/LINUX, desarrollado originalmente por Linus B. Torvalds a principios de la década de los noventa. Su estructura general es la típica de cualquier sistema UNIX: núcleo, intérprete de comandos, aplicaciones, procesadores de texto, servidores de red, bases de datos, aplicaciones multimedia y juegos.

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*. Los contenidos de una Distribución deben 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 apoyadas por una empresa comercial y esto tiene ciertas implicaciones por el modo en que se da apoyo. El apoyo para estas distribuciones procede de terceros y del acceso a listas de correo en Internet. Otras como *Ubuntu*, son elaboradas y respaldadas por empresas comerciales como Canonical y poseen gran aceptación por el usuario final.

## 1.3- La gran rivalidad.

GNU/LINUX es una familia de sistemas operativos LIBRES y Windows es una de las familias de sistemas operativos propietarios, en este caso, propiedad de Microsoft. Si existe un rival para Microsoft Windows casi seguro el más importante es GNU/LINUX, ya que está cada vez más, ganando lugares en el ramo de servidores, por el simple hecho de ser libre, más potente, configurable, seguro y estable. (3)

Aspectos a considerar	Software propietario	Software Libre
Acceso al código fuente	Prohibido por licencia	Si, garantizado
Corrección de errores en el programa por el cliente usuario	No	Si
Duplicación del software	Prohibido	Posible y recomendada
Libertad de competencia para el mantenimiento	No, depende del fabricante	Si, imposible limitarla
Posibilidad de examinar el código del producto	Prohibido salvo permiso del fabricante	Si
Venta de 2da mano	Prohibido	N/A
Respeto a estándares globales	En función del fabricante	En la mayoría de los casos

Adaptaciones al cliente	En función del fabricante	Disponible
<b>Virus, gusanos</b>	<b>Frecuentes</b>	<b>Muy infrecuentes</b>

Acevedo (2006) señala que para los proveedores del software las ventajas del Software Libre sobre el software propietario van mucho más allá de la parte técnica. Tanto los clientes consumidores de software como sus proveedores mejoran su actividad mediante el uso de Software Libre. Los proveedores pueden ofrecer tiempos de respuesta más bajos, mejor calidad, menos tiempo en I+D, planificaciones más exactas, ahorro en marketing (las aplicaciones libres suelen tener su propia Web, documentación, explicación de ventajas y son de libre descarga y uso), menos pruebas piloto y demostraciones, entre otras. Todo este ahorro se suele invertir en adaptación del software y soporte técnico al cliente.

No es necesario decir que en la mayoría de los casos, las aplicaciones libres reciben contribuciones por parte de las empresas que las usan y así se contribuye a la mejora global de éstas, contribuciones que provienen del uso comercial, de las personalizaciones y de la detección de errores.

Para los usuarios finales (clientes), las ventajas son también numerosas. De las anteriormente mencionadas, las relacionadas con la planificación y el tiempo de respuesta son ventajas para ambos, proveedor y cliente. Por otro lado el cliente tiene la libertad de probar el software, instalarlo, y, sobretodo, de decidir quien va a ofrecerle el soporte técnico. Este último es un punto clave por el que considerar migrar a Software Libre aplicaciones que dependen de una única empresa proveedora de software (ya sea una empresa local o una gran multinacional).

En el caso de Software Libre se puede contar con el respaldo y servicio de diversas distribuciones GNU/LINUX comerciales como RedHat, Suse o Mandriva, las cuales asumen la responsabilidad de corregir un programa si se presentan fallas o de emitir los parches o actualizaciones necesarias en el menor tiempo posible. Las actualizaciones pueden ser realizadas por programadores que no pertenecen al equipo original (de desarrollo) de acuerdo a sus intereses y gracias al acceso libre al código fuente.

Algunas distribuciones GNU/LINUX que son completamente libres (como Debian o Ubuntu), ofrecen un adecuado nivel de actualizaciones, sin embargo no existe un compromiso formal con el usuario para este soporte y para el caso de aplicaciones no críticas puede resultar una opción conveniente. Adicionalmente se abre para el mercado local también la posibilidad de ofrecer servicios de soporte y mantenimiento.

Tener en consideración que mientras las actualizaciones que elabora Microsoft son solamente para su sistema operativo Windows, en el caso de las distribuciones GNU/LINUX las actualizaciones abarcan, aparte del sistema operativo Linux, todos los programas que vienen en la distribución, como navegadores, paquetes de oficina, programas servidores, aplicaciones multimedia, utilitarios, etc., que pueden sumar cientos de programas, es por ello que no tiene sentido indicar que Windows es más seguro que una distribución GNU/LINUX al tener el primero menor cantidad de actualizaciones.

“Otra ventaja para el usuario final o cliente es la existencia de numerosas aplicaciones ya probadas y usadas por cientos o miles de usuarios. Y aún más importante es la posibilidad de descargarlas y usarlas antes de querer personalizarlas u obtener

soporte sobre ellas. Sin 30 días de prueba, sin "banners" publicitarios, simplemente descargar, instalar y usar." (5)

## **1.4- Panorama Mundial.**

Emitir una opinión sobre el uso de Software Libre requiere de una reflexión sobre diversos temas incluyendo el análisis técnico-económico, pues el Software Libre es considerado un movimiento social cuyo mayor impacto está en las tecnologías de información y comunicación, y a través de ellas en la sociedad en su conjunto.

Una vez que un producto de Software Libre ha empezado a circular, rápidamente está disponible a un costo muy bajo o sin costo alguno. Al mismo tiempo, su utilidad no decrece. Esto significa que el Software Libre se puede caracterizar como un bien público en lugar de un bien privado. Aunque realmente no lo es en ningún momento.

Puesto que el Software Libre permite el libre uso, modificación y redistribución, a menudo encuentra un hogar en los países del tercer mundo para los cuales el costo del software no libre es a veces prohibitivo. También es sencillo modificarlo localmente, lo que permite que sean posibles los esfuerzos de traducción a idiomas que no son necesariamente rentables comercialmente.

La mayoría del Software Libre se produce por equipos internacionales que cooperan a través de la libre asociación. Los equipos están típicamente compuestos por individuos con una amplia variedad de motivaciones. Existen muchas posturas acerca de la relación entre el Software Libre y el actual sistema económico capitalista:

- Algunos, consideran el Software Libre como un competidor del capitalismo, una forma de anarquismo práctico.
- Otros, como otra forma de competición en el mercado libre, y que el copyright es una restricción gubernamental sobre el mercado.
- No faltan los que consideran el Software Libre como una forma de cooperación en un modelo de mercado en una línea próxima al mutualismo.
- Hay quienes comparan el Software Libre a una economía del regalo, donde el valor de una persona está basado en lo que ésta da a los demás, así como un parentesco a la economía participativa. (6)

El Software Libre ha evolucionado y se ha consolidado en muchas partes del mundo teniendo un gran respaldo por parte de académicos, organizaciones educativas, grandes corporaciones, empresas, desarrolladores y usuarios de software. El tema ha trascendido del aspecto técnico para llegar a ser un tema estratégico en muchas organizaciones y un tema político en algunos países. *El Software Libre no es una moda, sino es, además de lo indicado, un modelo de negocio para una nueva industria de software basada en servicios, más que en productos.*

En muchos países hay antecedentes sobre de migración en entidades gubernamentales:

- Migración Software Libre en el gobierno de Ciudad de México DF (México).
- Municipalidad de Múnich. (Alemania).
- Accord-cadre avec le MENRT pour l'informatique libre (Francia)
- Grupo de Trabajo Europeo sobre el Software Libre. (Unión Europea)
- Proyecto Software Libre-Brasil (Brasil)

La introducción del Software Libre en entidades del Estado ha ido acompañada de diversos dispositivos o iniciativas legales como las siguientes:

- Francia - Proposition de Loi 117.
- Brasil - Lei (Pinheiro) – Recife - Proyecto Rio Grande do Sul.
- Argentina - Proyecto Diputado Marcelo Dragán.
- Dinamarca - Forslag til folketingsbeslutning om en strategi for udbredelse af open source-programmel i Danmark.
- Y otros tratados en la República Popular de China y Perú.

Uno de los casos más recientes y paradigmáticos de institución que ha decidido apostar de manera fuerte por el uso y la promoción de sistemas libres es la institución regional de Extremadura. La acción más emblemática hasta el momento ha sido la de elaborar una distribución de GNU/LINUX llamada LinEx, poniendo especial énfasis en la facilidad de instalación, y que cuenta con todas las herramientas que cualquier usuario doméstico medio utiliza habitualmente.

El primer fruto de este trabajo ha sido el ahorro de 30.050.605,2 euros en licencias de ordenadores para la enseñanza secundaria, que se han invertido en la compra de más equipos informáticos hasta el punto de lograr una ratio de dos alumnos por ordenador en esta etapa educativa.

En el caso de Chile, el Software Libre está avanzando lento pero seguro en las entidades gubernamentales. Son soluciones muy utilizadas los servidores GNU/LINUX (Web, FTP, Correo) y lenguajes como Java y PHP, bases de datos como Postgre, son una opción bastante utilizada.

En Venezuela se sigue con interés el proceso de migración al Software Libre. El decreto 3390 busca empujar a las instituciones gubernamentales del país a buscar alternativas libres migrando los sistemas, proyectos y servicios informáticos a esta nueva plataforma de desarrollo, instando a la Institución Pública Nacional a tomar cartas en el asunto. (7)

La globalización, y en especial la generalización del uso de Internet en el mundo desarrollado han facilitado el advenimiento de operadores globales en el mundo del software. Los mayores, Microsoft, HP, Oracle, IBM, Cisco, son corporaciones transnacionales de origen Estadounidense.

El Software Libre se constituye en una alternativa a las soluciones propietarias para la mayoría de los ámbitos públicos y privados. Este conjunto de soluciones informáticas generadas bajo distintas licencias, facilitan la reutilización de la experiencia (al estilo del conocimiento científico) y su uso generalizado y gratuito.

Actualmente existen numerosos programas distribuidos de manera libre ejecutándose en miles de máquinas. El auge de Internet ha favorecido claramente su extensión, al ser distribuidos de manera sencilla. Los programas, creados por personas altruistas y de manera desinteresada, son utilizados ya por miles de empresas y personas. Estas últimas se agrupan en comunidades con intereses comunes. (8)

## **1.5- Cuba**

Sin duda alguna, el uso del Software Libre es sustentable en Cuba a partir de las ventajas que tiene con respecto a los del tipo propietario. Por esto, su aplicación como

plataforma informática de trabajo adquiere una relevante significación que puede verse desde 4 ámbitos diferentes:

**POLITICO:** Desde un primer punto de vista, representa la no utilización de productos informáticos que demanden la autorización de sus propietarios (licencias) para su explotación. Es válido recordar que, en el presente Cuba se encuentra a merced de la empresa norteamericana Microsoft, que tiene la capacidad legal de reclamar a Cuba que no siga utilizando un sistema operativo de su propiedad, basado en leyes de propiedad industrial por las cuales también Cuba se rige; esto provocaría una interrupción inmediata del programa de informatización de la sociedad que como parte de la batalla de ideas está desarrollando el país, además pudiera implementarse una campaña de descrédito a la isla, abogando el uso de la piratería informática por parte de las instituciones estatales cubanas.

Desde un segundo punto de vista, el Software Libre representa la alternativa para los países pobres, y es por concepción, propiedad social, si se tiene en cuenta que una vez que comienza a circular rápidamente se encuentra disponible para todos los interesados sin costo alguno o en su defecto a muy bajo costo.

En tercer lugar, es desarrollado de forma colectiva y cooperativa, tanto en su creación como en su desarrollo, cuantitativa y cualitativamente, mostrando su carácter público y sus objetivos de beneficiar a toda la comunidad.

La posibilidad de usar, copiar, estudiar, modificar y redistribuir libremente el software como un bien social, que brinda esta plataforma, cumple los preceptos enunciados por la sociedad socialista cubana y está acorde con el tipo de economía socialista, donde el valor social está por encima de la ganancia.

**ECONÓMICO:** Su utilización no implica gastos adicionales por concepto de cambio de plataforma de software, por cuanto es operable en el mismo soporte de hardware con que cuenta el país.

La adquisición de cualquiera de sus distribuciones puede hacerse de forma gratuita, descargándolas directamente de Internet o en algunos casos a muy bajos precios, se garantiza su explotación con un mínimo de recursos, por cuanto no hay que pagar absolutamente nada por su utilización (no requiere de licencia de uso, las cuales son generalmente muy caras), distribución y/o modificación.

El uso del Software Libre desarrollado con Estándares Abiertos, fortalecerá la industria del software nacional, aumentando y fortaleciendo sus capacidades. Facilitará la reducción de la brecha social y tecnológica en el menor tiempo y costo posibles. Su uso en la Institución Pública y en los servicios públicos, facilitará la interoperabilidad de los sistemas de información del Estado, contribuyendo a dar respuestas rápidas y oportunas a los ciudadanos, mejorando la gobernabilidad.

**TECNOLÓGICO:** Permite su adaptación a los contextos de aplicación, al contar con su código fuente, lo cual garantiza un mayor porcentaje de efectividad, además de la corrección de sus errores de programación y la obtención de las actualizaciones y las nuevas versiones. (9)

Todas las mejoras que se realicen no tienen restricciones. De este modo, cualquier otra administración, empresa, institución o organismo se puede beneficiar de las mejoras introducidas.

Se fomenta la innovación tecnológica del país. Al disponer del código fuente de la aplicación, podemos realizar el desarrollo de mejoras, en vez de encargárselas a empresas de otros países que trabajan con sistemas de licencia propietaria. De este modo, contribuimos a la formación de profesionales en nuevas tecnologías y al desarrollo local bajo nuestros propios planes estratégicos.

Proceso de corrección de errores muy dinámico. Los usuarios del programa de todo el mundo, gracias a que disponen del código fuente del programa, pueden detectar los posibles errores, corregirlos, y contribuir con sus mejoras.

Más dificultad para introducir código malicioso, espía o de control remoto. Debido a que el código es revisado por muchos usuarios que pueden detectar posibles puertas traseras.

Por las razones detalladas anteriormente, el uso del Software Libre es, sin lugar a dudas, sustentable para Cuba y en ese sentido desde octubre del 2002, se puso en marcha una estrategia para alcanzar la independencia en el terreno del software, garantizando la seguridad informática y por sobre todas las cosas afianzando el uso de los principios del Software Libre, pues la negación de dichos preceptos constituiría el rechazo de los principios del socialismo y el comunismo.

Durante el año 2005 y principios del 2006, el país creó y consolidó el Grupo Técnico Nacional de Software Libre, lanzó una 1ra versión del Portal Cubano de Software Libre, realizó un Festival de Instalación y solicitó a la Universidad de las Ciencias Informáticas tomar cartas en el asunto destinándose una facultad al trabajo con este fin.

## ***1.6- Concluyendo esta parte.***

GNU/LINUX es un ejemplo claro del potencial de las comunidades para la creación del software. Al mismo tiempo es un software utilizado por los informáticos para realizar su trabajo. Lo que resalta el alcance en la sociedad en general de los programas de código libre. Por el momento, el impacto está bastante limitado a sectores educativos o puramente a entornos empresariales tecnológicos.

Cada vez son más las medidas que están tomando numerosas empresas e incluso gobiernos, para favorecer el uso de este tipo de software. Hoy existen empresas que han migrado sus sistemas a GNU/LINUX, y gobiernos que promueven iniciativas para el uso de Software Libre en su institución, lo que aumentará su difusión cada vez más. El Software Libre es un movimiento destinado a quedarse, y no hay duda de que impactará notablemente en el sector.

# **Parte 2**

# **CATEGORÍAS, LICENCIAS Y ESTÁNDARES.**

## 2- CATEGORÍAS, LICENCIAS Y ESTÁNDARES.

Un programa es software libre si se tiene:

- libertad para ejecutar el programa, con cualquier propósito.
- libertad para modificar el programa y adaptarlo a sus necesidades. (Para que esta libertad sea efectiva en la práctica, usted debe tener acceso al código fuente, porque modificar un programa sin disponer del código fuente es extraordinariamente dificultoso).
- libertad para redistribuir copias, tanto gratis como por un canon.
- libertad para distribuir versiones modificadas del programa, de tal manera que la comunidad pueda beneficiarse con sus mejoras.

**Estándares abiertos:** Especificaciones técnicas, publicadas y controladas por alguna organización que se encarga de su desarrollo, las cuales han sido aceptadas por la industria, estando a disposición de cualquier usuario para ser implementadas en un software libre u otro, promoviendo la competitividad, interoperabilidad o flexibilidad.

### 2.1- Categorías de Software Libre.

#### 2.1.1- Software de código abierto (open source).

En 1998, una parte de la comunidad decidió dejar de usar el término “free software” (software libre) y usar “open source software” (software de código abierto), con el propósito de evitar la confusión de “free” con “gratis”. Otros, sin embargo, apuntaban a apartar el espíritu de principios que ha motivado el movimiento por el software libre y el proyecto GNU, y resultar así atractivos a los ejecutivos y usuarios comerciales. Open source se centra en el potencial de realización de software de alta calidad, pero esquiva las ideas de libertad, comunidad y principio.

“Free software” y “open source” describen la misma categoría de software, más o menos, pero dicen diferentes cosas acerca del software y acerca de los valores. El proyecto GNU continúa utilizando el término “free software” para expresar la idea de la libertad, donde solamente la tecnología es lo importante.

#### 2.1.2- Software de dominio público.

El software de dominio público es un tipo de software que no está protegido con copyright. Es un caso especial de software libre no protegido con copyleft, que significa que algunas copias o versiones modificadas no pueden ser libres completamente. “Dominio público” es un término legal y significa de manera precisa “sin copyright”.

#### 2.1.3- Software protegido con copyleft.

El software protegido con copyleft es software libre, cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando estos redistribuyen o modifican el software. Lo antes expuesto significa que cada copia del

software, aún si ha sido modificado, debe ser libre. El proyecto GNU protege mediante copyleft casi todo el software que producen, con el propósito de dar a cada usuario las libertades que el término "software libre" implica. Copyleft es un concepto general. Para proteger actualmente un programa con copyleft se necesita usar un conjunto específico de términos de distribución. Hay muchas maneras posibles de hacerlo.

### **2.1.4- Copyleft y la GNU GPL.**

El copyleft usa la ley de copyright pero le da un giro para servir a lo opuesto de su propósito usual. En lugar de ser un medio de privatizar el software, se transforma en un medio de mantener libre al software. La idea central del copyleft es dar a cualquiera el permiso para correr el programa, copiarlo, modificarlo y redistribuir versiones modificadas, pero no se da permiso para agregar restricciones propias. De esta manera, las libertades cruciales que definen al software libre quedan garantizadas para cualquiera que tenga una copia transformándose en derechos inalienables.

Para que el copyleft sea efectivo, las versiones modificadas deben ser también libres. Esto asegura que todo trabajo basado en GNU quedará disponible para la comunidad si se publica.

Cualquier cosa agregada o combinada con un programa bajo copyleft debe ser tal que la versión combinada total sea también libre y bajo copyleft.

La implementación específica de copyleft para la mayoría del software GNU es la Licencia Pública General de GNU (GNU General Public License) o GPL GNU para abreviar. Los manuales GNU también están bajo copyleft, pero se utiliza un copyleft mucho más simple, porque no es necesaria la complejidad de la GPL GNU para los manuales.

Si bien la licencia GPL ofrece grandes beneficios, hay algunas veces en la que ofrece ciertas restricciones. Un ejemplo es que un software que utiliza algún componente GPL, debe sí y solo sí ser licenciado bajo la misma, es decir no se pueden utilizar partes o bibliotecas de software GPL en un software propietario o distribuido bajo otra licencia.

Estas restricciones traen algunos problemas. Por ejemplo si una empresa comercial desea utilizar únicamente una biblioteca GPL pequeña dentro de su software, estaría obligada a distribuir todo su software bajo GPL, lo cual posiblemente no decida hacer y para algunos casos como el de bibliotecas de propósitos generales, esto tampoco ayuda a la mejora de la propia biblioteca, ya que no sería elegida por ejemplo, para convertirse en un estándar. Por esto apareció la licencia LGPL, en un primer momento llamada Library GPL en referencia a que fue especialmente utilizada para bibliotecas, pero luego se popularizó y comenzó a utilizarse inclusive en muchos programas completos, debido a sus beneficios comerciales (permite utilizarse junto a software no libre) y cambió su nombre a Lesser GPL que significa GPL menos restrictiva.

La licencia pública general limitada de GNU, o GNU Lesser General Public License (GNU LGPL), es una licencia de software creada por la Free Software Foundation (Fundación del Software Libre). Los contratos de licencia de la mayor parte del software están diseñados para jugar con su libertad de compartir y modificar dicho software. En contraste, la "GNU, General Public License" pretende garantizar su libertad de compartir y modificar el software "libre", esto es para asegurar que el software es libre para todos sus usuarios.

Esta licencia pública general se aplica a la mayoría del software de la Free Software Foundation (FSF) y a cualquier otro programa de software cuyos autores así lo establecen. Algunos otros programas de software de la FSF están cubiertos por la LGPL, la cual puede aplicarse a cualquier programa o trabajo que contenga una nota puesta por el propietario de los derechos del trabajo estableciendo que su trabajo puede ser distribuido bajo los términos de esta GPL. El "Programa", utilizado en lo subsecuente, se refiere a cualquier programa o trabajo original, y el "trabajo basado en el Programa" significa ya sea el Programa o cualquier trabajo derivado del mismo bajo la ley de derechos de autor: es decir, un trabajo que contenga el Programa o alguna porción de él, ya sea íntegra o con modificaciones o traducciones a otros idiomas.

Otras actividades que no sean copia, distribución o modificación si están cubiertas en esta licencia y están fuera de su alcance. El acto de ejecutar el programa no está restringido, y la salida de información del programa está cubierto sólo si su contenido constituye un trabajo basado en el Programa (es independiente de si fue resultado de ejecutar el programa). Si esto es cierto o no depende de la función del programa.

### **2.1.5- Software libre no protegido con copyleft.**

El software libre no protegido con copyleft viene desde el autor con autorización para redistribuir y modificar así como para añadirle restricciones adicionales.

Si un programa es libre pero no protegido con copyleft, entonces algunas copias o versiones modificadas pueden no ser libres completamente. Una compañía de software puede compilar el programa, con o sin modificaciones, y distribuir el archivo ejecutable como un producto propietario de software. El sistema X Window ilustra esto. El consorcio X libera X11 con términos de distribución que lo hace software libre no protegido con copyleft. Si usted lo desea puede obtener una copia que tenga esos términos de distribución y es libre. Sin embargo, hay versiones no libres también y hay estaciones de trabajo populares y tarjetas gráficas para PCs para las cuales versiones no libres son las únicas que funcionan. Si usted está usando este hardware, X11 no es software libre para usted.

### **2.1.6- Software abarcado por GPL.**

La GPL (General Public License/Licencia Pública General) de GNU es un conjunto específico de términos de distribución para proteger con copyleft a un programa. El proyecto GNU la utiliza como los términos de distribución para la mayoría del software GNU.

### **2.1.7- Software GNU.**

El software GNU es liberado bajo el auspicio del Proyecto GNU. Todo el software GNU debe ser software libre, aunque existen un grupo de estos software no están protegidos con copyleft, siendo estos la minoría.

Muchas partes de software GNU es escrito por el personal de la Free Software Foundation, pero la mayoría del software GNU es aportada por voluntarios. Parte del software aportado está protegido con copyright por la fundación, otra parte, está protegido con copyright por los programadores voluntarios que las escribieron.

### **2.1.8- Software semilibre.**

El software semilibre no es libre, pero viene con autorización para usar, copiar, distribuir y modificar (incluyendo la distribución de versiones modificadas) sin fines de lucro. PGP (Pretty Good Privacy) es un ejemplo de un programa semilibre.

El software semilibre es una apuesta un poco más visionaria que el software propietario, pero aún plantea problemas y además no puede usarse en un sistema operativo libre. Esto obedece a que los términos de distribución para el sistema operativo libre como un todo, es la conjunción de los términos de distribución de todos los programas en él.

### **2.1.9- Software propietario.**

El término “software propietario” es software que no es libre ni semilibre. Su uso, redistribución o modificación está prohibida, requiere que usted solicite autorización o está tan restringida que no pueda hacerla libre de un modo efectivo.

### **2.1.10- Freeware.**

La palabra “freeware” no tiene una definición clara aceptada, pero es usada comúnmente para paquetes que permiten la redistribución pero no la modificación (y su código fuente no está disponible). Estos paquetes no son software libre.

### **2.1.11- Shareware.**

El vocablo significa literalmente programa compartido e indica que cualquiera pueda descargar el programa y empezar a emplearlo sin desembolso previo durante un período de prueba. Esto no significa que sea de libre uso o de empleo gratuito. La licencia de uso indica con claridad en cada caso los términos de empleo, así como la cantidad que debe ser abonada en caso de encontrarse de utilidad el programa.

El sistema shareware se utiliza a menudo como medio para distribuir versiones de prueba con un coste mínimo. Las versiones de prueba, en general tienen algún tipo de limitación. En algunos casos, algunas funciones no están disponibles; en otros, el programa solo admite una cierta cantidad, reducida de datos. En su versión más popular, el programa tiene toda su funcionalidad, pero solo es operativo durante 30 días tras su instalación. Al cabo de estos, unos programas dejan de funcionar y recuerdan que deben ser desinstalados de la computadora o pagados. Algunos programas simplemente recuerdan cada vez que se ejecutan que el período de prueba ha terminado, pero siguen operativos. (10)

El shareware no es software libre, ni siquiera semilibre, debido a las razones siguientes:

- Para la mayoría del shareware, el código fuente no está disponible; de esta manera, usted no puede modificar el programa en absoluto.
- El shareware no viene con autorización para hacer una copia e instalarlo sin pagar una cantidad por licencia, ni aún para particulares involucrados en actividades sin ánimo de lucro. (En la práctica, la gente a menudo hace caso omiso a los términos de distribución y lo copian e instalan de todas formas,

pero estos no lo permiten).

RAR y WinRAR son programas shareware, esto significa que se *pueden probar gratuitamente durante 40 días*, pasado este período de prueba deberá comprarse una licencia o desinstalarlos de la computadora. La licencia es *válida de por vida* y da derecho a *todas las actualizaciones* que vayan saliendo del programa de forma *gratuita* y sin ningún coste ni cuota extra de mantenimiento.

Hay también software shareware que dejan de funcionar después de un periodo de prueba, los llamados Try Out. [11]

### **2.1.12- Software comercial.**

El término “software comercial” se utiliza para el software que está siendo desarrollado por una entidad que tiene la intención de generar ganancias económicas mediante el uso del software. Comercial y propietario no son equivalentes. La mayoría del software comercial es propietario, o sea, es propiedad de una empresa que lo desarrolla con el interés de obtener ganancias por su uso y comercialización. No obstante, hay software libre con intereses comerciales, aunque por lo general el software libre es no comercial.

De manera general el software comercial se caracteriza porque:

- Tiene licencias, las cuales están limitadas por usuarios y son pagas. Estas licencias restringen las libertades de los usuarios a usar, modificar, copiar y distribuir el software.
- El desarrollo, programación y actualización de este software solo lo hace la empresa que tiene los derechos. Como sucede con los productos Microsoft (Windows, Office, etc.). Estos ocasiona retrasos tecnológicos y poca creatividad en los productos.
- En el software comercial se suele esconder y mezquinar los avances y descubrimientos tecnológicos entre las empresas que lo desarrollan.
- El futuro del software que compro el usuario solo depende de una empresa comercial.
- Muchas veces con estrategias comerciales se suele hacer que los usuarios actualicen su software comercial, sin que exista una necesidad verdadera de ello, consiguiendo de esta forma hacer que el usuario invierta en nuevas licencias, la mayoría de las veces innecesarias. [12]

Esta modalidad de software tiene muy buenos exponentes, tales como:

- MySQL (My Structured Query Language) que es un sistema de administración relacional de bases de datos que almacena la información en tablas separadas, en vez de colocar toda la información en un único archivo. Está continuamente en desarrollo, es software libre. Posee gran velocidad y flexibilidad. Su conectividad y robustez hacen de él un buen sistema gestor de bases de datos. [13]
- Zimbra, que es un cliente/servidor de correo y calendario, al estilo del Yahoo!

Mail en cuanto contenido y al estilo de Gmail en cuanto a velocidad. Para hacer un interfaz tan rápido han usado, al igual que Gmail, del lenguaje AJAX que almacena parte de la página en el cliente, por lo que gran parte de la comunicación entre el cliente y el servidor se reduce inversamente proporcional a la velocidad. No sólo se puede decir que Zimbra es uno de los mejores clientes web/servidores de correo y calendario, sino que además es código libre, gratuito y descargable, tanto su cliente como el servidor. Ha sido desarrollado en Java, complementado con AJAX. Soporta acceso POP, acceso IMAP, entre otros; e incluye protección anti-spam y antivirus. [14]

## **2.2- Licencias de Software Libre.**

### **2.2.1- Software bajo licencia BSD.**

Berkeley Software Distribution (BSD). Es la licencia original de una distribución de Software que acabó convirtiéndose en un derivativo de UNIX. Ha tenido dos formas fundamentales: la clásica (con la cláusula de publicidad) y la actual (sin esa cláusula desde julio del 99). [15]

Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre. El autor, bajo esta licencia, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación. [16]

### **2.2.2- Software bajo licencia Apache.**

Creada por la Apache Software Foundation (ASF). La licencia Apache (con versiones 1.0, 1.1 y 2.0) requiere la conservación del aviso de copyright y el disclaimer, pero no es una licencia copyleft, ya que permite el uso y distribución del código fuente para software libre y software propietario. [17]

La licencia Apache es una descendiente de las licencias BSD por lo que no es GPL. Esta licencia permite hacer todo tipo de cosas con el código fuente (incluyendo productos propietarios) siempre que se les reconozca su trabajo. Todo el software producido por la ASF o cualquiera de sus proyectos está desarrollado bajo los términos de esta licencia. [18]

La fundación Apache (Apache Foundation) recibe dinero de todas las empresas que venden productos basados en apache (HP, IBM, Oracle, Sun). Con este dinero se paga a los desarrolladores. El producto se regala a la gente...y esas empresas venden sus propias versiones "mejoradas" del producto.

Esas empresas ganan debido a que tienen acceso a la tecnología y a que venden las versiones "empresariales" de los productos, la inversión se reparte y se cuenta con la ventaja de un número mayor de colaboradores. [19]

### **2.2.3- Software bajo licencia Mozilla.**

Mozilla Public License (MPL), es una licencia de código abierto y software libre. Fue

desarrollada originalmente por Netscape Communications Corporation –una división de la compañía 'América Online'–, y más tarde su control fue traspasado a la 'Fundación Mozilla'.

Cumple con la definición de software de código abierto y con las cuatro libertades del software libre enunciadas por la Free Software Foundation (FSF). Sin embargo la MPL deja abierto el camino a una posible reutilización no libre del software, si el usuario así lo desea, sin restringir la reutilización del código ni el relicenciamiento bajo la misma licencia.

Aunque el uso principal de la MPL es servir como licencia de control para el navegador Mozilla y el software relacionado con él (el navegador Firefox o el cliente de correo Mozilla Thunderbird, por ejemplo), esta licencia es ampliamente utilizada por desarrolladores y programadores que quieren liberar su código. [20]

### **2.2.4- Software bajo licencia GPL 3.**

Lo esencial de la GPL no ha cambiado respecto a GPL2: libertad de usar el programa, libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades, libertad de distribuir copias, libertad de mejorar el programa y hacer públicas las mejoras. [21]

En esta nueva versión se intentan solucionar una serie de problemas a los que se ha tenido que enfrentar la comunidad del software libre, entre ellos el tema de patentes de software y el conocido como tivolización. Con GPLv3, si una empresa utiliza software con esta licencia ofrece permiso gratuito para utilizar las patentes que lo cubren al resto de usuarios. Garantiza además que los acuerdos de protección de patentes como el que hicieron Microsoft y Linpire o Microsoft y Novell se extiendan al resto de usuarios de ese software.

Finalmente, el tema de la tivolización está relacionado con el DRM y su nombre viene porque la empresa Tivo distribuye un hardware con Linux, cumpliendo la licencia, pero el problema para el usuario es que no puede modificar el software y ejecutarlo a causa del DRM que se utiliza. La GPLv3 quiere evitar que se puedan producir este tipo de situaciones. [22]

## **2.3- Nociones de Estándares Informáticos.**

Una de las principales herramientas que han permitido a las distintas herramientas informáticas poder interactuar y así proporcionar una experiencia satisfactoria al usuario han sido los estándares. Los estándares, o mejor dicho, parte de ellos como se comprobará en adelante, han sido la herramienta base de la interoperabilidad informática. Son los que han permitido definir cómo interactuaran los miles o millones de componentes informáticos que existen.

Sin embargo, estándares existen de muchos tipos y según de cuál de ellos se esté hablando, se estarán garantizando unas funcionalidades y unas capacidades de interoperabilidad técnica distintas. Así, los estándares se pueden clasificar en función de diversas características. Las dos principales probablemente, de cara a las implicaciones que tienen de cara a su uso son cómo de abiertos/cerrados y permisivos/exclusivos son, y qué carácter legal tienen. También es interesante observar qué organismo ha emitido y es responsable del estándar, así como su ámbito



industrial.

### **2.3.3- Estándar cerrado.**

La especificación del estándar ha sido hecha pública, pero sin embargo, existen determinadas restricciones legales (principalmente patentes, pero también derechos de autor, marcas, etc.) que impiden que se pueda implementar el estándar libremente por parte de aquellos que no lo desarrollaron o adquirieron sus derechos.

Así, los términos de la licencia de implementación de las especificaciones no son públicos ni comunes para todos los posibles agentes del mercado interesados en implementar el estándar, esto es, en crear aplicaciones o herramientas informáticas que sigan cumplan con lo definido en las especificaciones del estándar. De esta forma, cada implementador se ve forzado a llegar a un acuerdo particular con cada uno de los propietarios de cada una de las exclusividades legales que forman el estándar. Esto puede imposibilitar a muchos desarrolladores (empresas o personas) llevar a cabo una implementación compatible con el estándar, y, en todo caso, puede discriminar a unos frente a otros en los términos obtenidos para la licencia. Generalmente este tipo de estándares quedan delimitados a un círculo pequeño de agentes que son los que pasan a controlar el mercado que genera el estándar.

Como cuerpos de estandarización que pueden proveer este tipo de estándares se puede encontrar a ECMA, una asociación industrial europea formada por muchas de las mayores multinacionales informáticas, y que en su proceso de estandarización no provee las suficientes garantías para que los miembros del comité técnico de estandarización en cuestión desvelen las patentes y otras restricciones a las que está sometido el estándar en definición o ratificación.

Los estándares cerrados pues, atan a determinados fabricantes (o incluso a un único fabricante) y por tanto discriminan a la población y al resto de agentes del mercado en función de las buenas o malas relaciones que tengan con los propietarios de las distintas partes del estándar.

Sólo podrán relacionarse con las aplicaciones que sigan esos estándares aquellos que sean clientes de los fabricantes que hayan llegado a acuerdos con los dueños del estándar, así como con los clientes de los propios dueños. El documento de especificación, es público, pero no por ello tiene que ser gratuito, pudiendo tener un coste simbólico.

### **2.3.4- Estándar RAND.**

Un estándar RAND es aquel estándar cuya especificación ha sido normalizada y es pública, y que ha sido licenciada bajo unos términos comunes para todo el mercado. Las patentes y otras posibles restricciones legales a las que esté sometido el estándar parcial o totalmente habrán sido hechos públicos durante el proceso de estandarización. Esto incluye al menos aquellos que los miembros del comité de estandarización tengan en posesión, no así aquellos que sean externos o que no cuenten con. Generalmente la forma "RAND" es el mínimo de apertura e inclusividad exigido por muchos de los principales organismos de estandarización (por ejemplo, ISO, IEC, OASIS, etc.)

El término RAND proviene del inglés "*Reasonable and Non Discriminatory*", sin embargo, y como se constata incluso mediante importantes pleitos en curso, dicho

término muy al contrario frecuentemente implica términos de licenciamiento que son poco razonables y muy discriminatorios.

El hecho de que la licencia sea común para cualquier implementador, no significa que la propia licencia no discrimine a partes del mercado o que los costes de licenciamiento no sean abusivos. Así, es frecuente que licencias RAND discriminen a modelos de desarrollo como los de código abierto, pues muchas veces obligan a que las implementaciones del estándar oculten el código (algo imposible para estos modos de desarrollo). También es común que no permitan la libre redistribución del software por parte de los usuarios del mismo, con lo que resultan discriminados los modelos de libre distribución como el software libre que en su definición contienen el derecho del usuario a distribuir libremente el software libre que recibe.

Por otro lado, al igual que los anteriores tipos de estándares, el documento de especificación, es público, pero no por ello tiene que ser gratuito, pudiendo tener un coste simbólico.

### **2.3.5- Estándar abierto.**

Un estándar abierto ha de disponer su especificación de forma pública (aunque quizá sujeta a algún pago simbólico en concepto de derechos de autor del documento en sí), el estándar ha de ser inclusivo y haber sido desarrollado y estar mantenido en un proceso de estandarización abierto. Todo el que esté interesado podrá implementarlo sin ninguna restricción, ni pago, si sujeto a derecho de exclusión alguna. Las licencias de los posibles propietarios del estándar o sus partes han de conceder esos derechos de forma gratuita y sin condición alguna a todos los agentes interesados en su implementación, independientemente de su modelo de desarrollo, situación geopolítica, grado de cumplimiento con la especificación, etc. En otras palabras estas condiciones equivalen a la expresión “libres de regalías”.

Los descritos son los mínimos términos de licenciamiento requeridos por ejemplo por cuerpos de estandarización tan importantes como W3C, el responsable de todos los formatos y protocolos de la web. Por supuesto, todos los otros cuerpos de estandarización aceptan estos términos de licenciamiento de estándar abierto, pues sobrepasan los mínimos requeridos por los mismos en cuanto a liberación de exclusividades.

Para mayor precisión, el Marco Europeo de Interoperabilidad, documento oficial emitido por la Comisión Europea en el año 2004, define los estándares abiertos como aquellos cuya especificación y sus documentos de apoyo cumplen como mínimo las siguientes condiciones:

- El estándar es adoptado y será mantenido por una organización sin ánimo de lucro, y sus sucesivos desarrollos se producen bajo la base de un procedimiento de toma de decisiones abierto disponible a todas las partes interesadas (consenso o decisión mayoritaria etc.)
- El estándar ha sido publicado y el documento de especificación del estándar está disponible ya gratuitamente o a coste simbólico. Debe permitirse a cualquiera copiarlo, distribuirlo y usarlo sin coste alguno o a un coste nominal.
- La propiedad intelectual del estándar p. ejem. patentes posiblemente presentes del (o de partes del) estándar se dispone irrevocablemente bajo una base libre de regalías.
- No hay limitaciones respecto al uso del estándar.



el caso de organismos como OASIS, W3C, ECMA, etc. [23]

# **Parte 3**

# **PENSANDO EN EL**

# **FUTURO.**

### **3- PENSANDO EN EL FUTURO.**

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

Muchas de las aplicaciones software libre funcionarán con sistemas operativos propietarios y esto nos brinda la oportunidad de introducir estas aplicaciones sin tener que cambiar totalmente el entorno. Por ejemplo Open Office (Suite Ofimática) y Firefox (Navegador Web) funcionarán con Windows y así pueden utilizarse en sustitución de Office, e Internet Explorer respectivamente. 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.

Muchas guías coinciden en afirmar que los primeros cambios son los que no afecten a la comunidad de usuarios. Eso quiere decir que estos se harán en el servidor, proporcionando la plataforma para la posterior introducción de los cambios en el lado del cliente, que 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 herramientas de software libre equivalentes 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 software libre puro, pero que permiten la coexistencia de los antiguos sistemas propietarios y el software libre. El uso temprano de estas puede ser muy eficaz.

#### **3.1- Primeras transformaciones.**

Evitar hacer cosas ahora que puedan dificultar la migración en el futuro, sería genial. Veamos:

1- Insistir en que los desarrollos web hechos tanto internamente como por empresas contratadas para ello, produzcan un contenido que se pueda visualizar en todos los navegadores actuales de la web, en particular los navegadores de software libre. Esta sería una buena práctica en cualquier caso ya que no debería requerirse software específico para visualizar su contenido. Hay herramientas como Weblint y otras más recientes, 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 la persona que lo esté leyendo y que se imprima cualquier otra cosa.

3- Insistir en el uso de formatos de archivos abiertos y estándar, como PostScript y

PDF. Hay cierta discusión sobre si PostScript 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 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. [24]

Office Open XML (también llamado OOXML u Open XML) es un formato de documento electrónico creado y desarrollado por Microsoft. Las especificaciones de este formato han sido desarrolladas por Microsoft para suceder a sus formatos binarios de archivo, y cedido a organismos de estandarización como ECMA e ISO. [25]

Pero no es abierto: pues para que un formato pueda ser considerado abierto, ha de estar libre de regalía o condición alguna por patentes, y no puede ser controlado por una única empresa (...que además en este caso cuenta con un amplio historial de amenazas a sus competidores mediante patentes de software). Por otro lado, el formato en cuestión debe ser multiplataforma por naturaleza. Office OpenXML no cumple ninguna de esas condiciones.

No es estándar: pues un estándar debe estar totalmente documentado públicamente. Además, su proceso de estandarización (ECMA) debería haber garantizado que todas las patentes de los proponentes del mismo fueran desveladas y licenciadas como mínimo en términos RAND. Finalmente, es necesario que una propuesta de estándar ISO cumpla y no contradiga los estándares ISO ya preexistentes para no obligar a "reinventar la rueda". Office OpenXML no cumple ninguna de esas condiciones.

Y no es XML: pues para que un formato de representación de información pueda ser considerado XML, éste debe plasmar íntegramente dicha información en estructuras XML. Para que algo se denomine XML no basta con que simplemente utilice etiquetas XML cuando en ellas guarda información en formatos binarios, con códigos de control e incluso dependiente de plataformas concretas. En resumen, debe validar el estándar XML. Office OpenXML no cumple estas condiciones. [26]

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. 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 2003 en lugar de Word 2007. Esto aumentará la posibilidad de que las aplicaciones de software libre puedan participar.

5- Desarrollar sistemas basados en por lo menos un modelo de tres niveles 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 de software libre. 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.

6- 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. Evitar lenguajes y APIs de arquitecturas específicas. Evitar la construcción de aplicaciones que requieran la presencia de otras aplicaciones propietarias.

7- Apartar a los usuarios de lectores de correo propietarios que usen formatos de buzón propietarios y se comuniquen con servidores que usen 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.

8- 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 del tema del 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. [27]

### **3.2- Las reacciones.**

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

El uso del software libre 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 software libre porque es nuevo para ellas.

Habrán usuarios que son más curiosos por naturaleza, que pueden sentirse felices de conocer cosas nuevas y son ellos los que deberían probar el software libre en primer lugar. Hasta ahora la experiencia indica que una vez que la gente vence sus reservas encuentra que el este software 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 software libre 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.

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 software libre sea de uso generalizado, las administraciones se pueden encontrar con el problema descrito con cierta frecuencia.

La gente que conoce los sistemas y configuraciones existentes tiene un cierto poder y podrían sentirse bastante reacios a perderlo si el entorno software libre 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. [28]

Lo importante es continuar.

### ***3.3- El proceso ideal.***

De manera general, el proceso de migración a Software Libre ideal debe consistir en los siguientes puntos, agrupados en el preciso número de etapas. Algunos de ellos pueden hacerse en paralelo en dependencia de las disponibilidades de las que se disponga a la hora de realizar el proceso.

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 tendrá que ser suficiente para permitir por lo menos la construcción de pilotos representativos, permitiendo 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 Libre como la arquitectura básica, junto con las diferentes opciones y posibilidades disponibles.** Esto significa que hay que formar al personal, contratar personal o recurrir a consultores. Lo que implicará algunos costos 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 costo alguno.
3. **La migración es una oportunidad de revisar la arquitectura de base así como el software de aplicaciones.** La arquitectura que se recomiende se debe basar en el control centralizado y debe tener ciertas ventajas. Puede haber ciertos costos al hacer el cambio, y hay que tenerlos en cuenta.
4. **Es muy importante entender bien en qué consiste el SWL.** Hay algunos aspectos que hay que tener en cuenta antes de tomar alguna

decisión:

- Hay que tener claro cuáles son las implicaciones de las licencias para SWL especialmente si se considera que la institución va a distribuir los cambios de software.
- Cuando hay varias opciones para una función (por ejemplo, hay por lo menos tres buenas hojas de cálculo de SWL) los ejecutores del proceso han de entender los pros y los contras de cada producto.
- 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 una adaptación del software para que satisfaga necesidades concretas. Todas estas diferencias han de ser valoradas antes de hacer cualquier elección.
- Los ejecutores 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 internacionales 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 costo total de propiedad para un caso concreto de negocio.
6. **Elaborar un caso detallado de migración,** que se basará en los datos recogidos y que consistirá en los siguientes puntos:
  - El costo del entorno existente en un período de tiempo razonable.
  - El costo de entornos alternativos y el costo de la migración a cada uno de ellos en el mismo período.
  - Los puntos fuertes y débiles del entorno actual y las distintas alternativas.
7. **Consultar a los usuarios.** Explicar las razones que hay detrás de la migración y cómo les afectará.
  - Estudiar sus preocupaciones con seriedad y permitirles que practiquen con la tecnología, sin pérdida de tiempo. Cuanto antes se impliquen los usuarios mejor será.
  - Habilitar un espacio 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. **Comenzar con proyectos pilotos a pequeña escala, de preferencia en un entorno auto-contenido con pocos usuarios.**

Esto facilitará, entre otras cosas:

- Datos más ajustados de modelos de costo total de propiedad.
- La reacción de los usuarios, que se puede emplear para facilitar la introducción a otros sistemas.
- 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:

- **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 pasando constantemente 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 instituciones. Las migraciones “big bang” tienen tantas variantes que controlar, que casi siempre fallan. Y si lo hacen, no parece probable que sea por un fallo del SWL, sino de la gestión.

- **Transición 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 las PC al mismo tiempo, reemplazando las máquinas en un grupo y luego instalando las sustituidas en lugar de las viejas máquinas del siguiente grupo.
- **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 instituciones. Pero sí puede ser una buena manera de ejecutar los proyectos pilotos.

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 Institución.** 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 detectar, 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. (29)

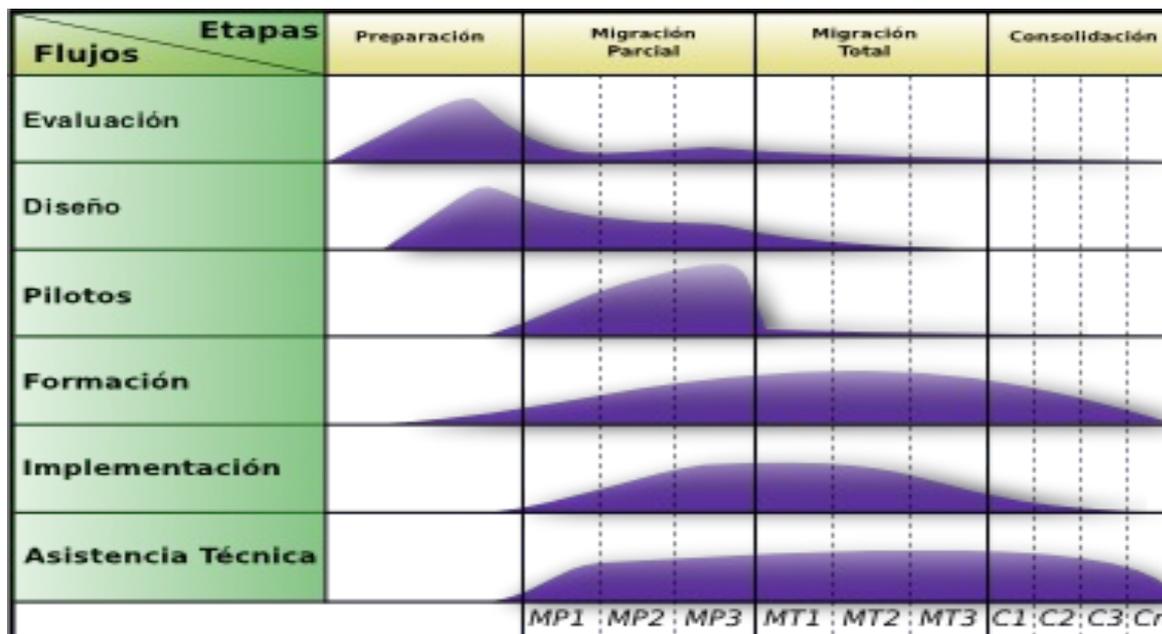
Ya tocados de cerca estos puntos estamos cerca de poder comenzar la migración...antes, echemos un vistazo a la metodología a seguir.

# **Parte 4**

# **METODOLOGÍA 2008.**

## 4- METODOLOGÍA 2008.

La metodología para la migración a Software Libre contempla 4 Etapas o Fases y 6 Flujos de Trabajo. Se describe como flujo de trabajo a la secuencia de acciones, actividades o tareas utilizadas para la ejecución de un proceso, incluyendo el seguimiento del estado de cada una de sus etapas y la aportación de las herramientas necesarias para gestionarlo.



### ETAPAS.

**Preparación:** Etapa en la que se realizarán las tareas de recopilación de datos y se lanzará una primera versión de la guía de migración.

**Migración Parcial:** Etapa en la que se realizarán las pruebas y se validará la propuesta a pequeña escala, además de que tendrá gran actividad de trabajo.

**Migración Total:** Cada vez que se ejecute una iteración de esta fase la cantidad de FLOSS irá en aumento, será la etapa que marcará el fin del software privativo.

**Consolidación:** Etapa que constituirá el soporte al proceso de migración, será el apoyo e indicará los niveles de éxito o fracaso de la Migración a Software Libre.

### FLUJOS DE TRABAJO.

**Evaluación:** Hacer una evaluación de todos los procesos, tecnología y personal y adaptarlas al entorno actual.

**Diseño:** Diseñar un plan de migración conforme a las necesidades, tomando como partida el resultado anterior.

**Pilotos:** Poner en marcha el plan en un ambiente real de pruebas.

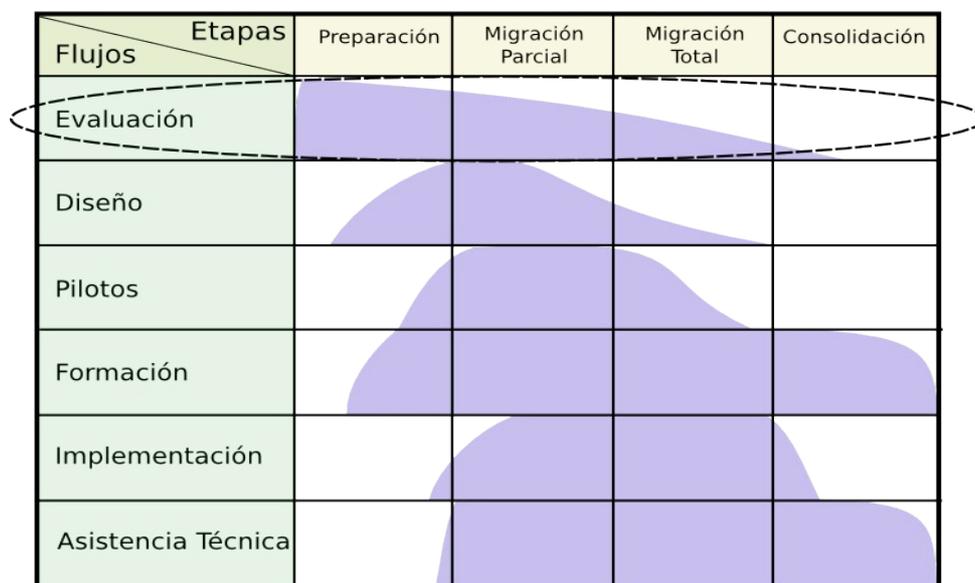
**Formación:** Formación del personal y certificación del mismo por niveles de usuarios.

**Implementación:** Instalación y migración definitiva de servicios y estaciones de trabajo a Software Libre.

**Asistencia y soporte técnico:** Brindará atención y soporte a las infraestructuras, servicios instalados y al personal. (30)

## 4.1- Descripción de los flujos de trabajo.

### 4.1.1- Flujo de trabajo: Evaluación.



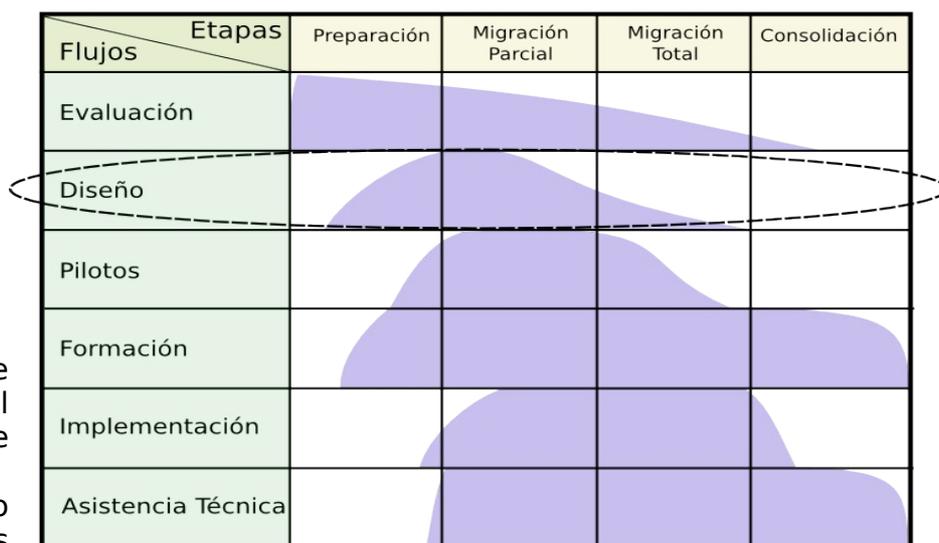
La

evaluación es donde se realiza una valoración de todos los procesos y tecnologías presentes. Es de suma importancia pues en él, se definen elementos que serán la base de la futura migración. Este flujo constituye un hito fundamental en el proceso y se realiza una sola vez, teniendo su mayor impacto en esta etapa.

Para este flujo se definen un conjunto importante de tareas:

- Evaluar los tipos de usuarios que participan en el proceso y clasificar los mismos según el grado de impacto, para definir prioridades durante los procesos de formación y concientización, y ver cuales pueden ser participes directos o no del proceso.
- Analizar el estado actual de los sistemas informáticos de la institución en cuanto a software y hardware. Para esta tarea puede ser de utilidad, debido a la magnitud de la misma, apoyarse en aplicaciones para automatizar el proceso como pueden ser OCS Inventory.
- Analizar el estado actual de los servicios que brinda la institución y determinar el grado de criticidad de los mismos para definir prioridades durante el proceso de cambio tecnológico.
- Evaluar los distintos escenarios posibles que se pueden seguir para acometer la migración.
- Evaluar las potenciales soluciones de migración disponibles para los sistemas informáticos presentes y definir los más adecuados en cada caso, así como, definir cuales serán necesarios desarrollar sus posibilidades reales de desarrollo.
- Hacer una evaluación de los mecanismos más adecuados que faciliten el soporte y la asistencia técnica dentro de estructura de la institución.
- Cuantificar las herramientas privativas a migrar.
- Hacer una evaluación del costo de la migración.

## 4.1.2- Flujo de trabajo: Diseño.



En este, se definirá el plan de migración de acuerdo a las

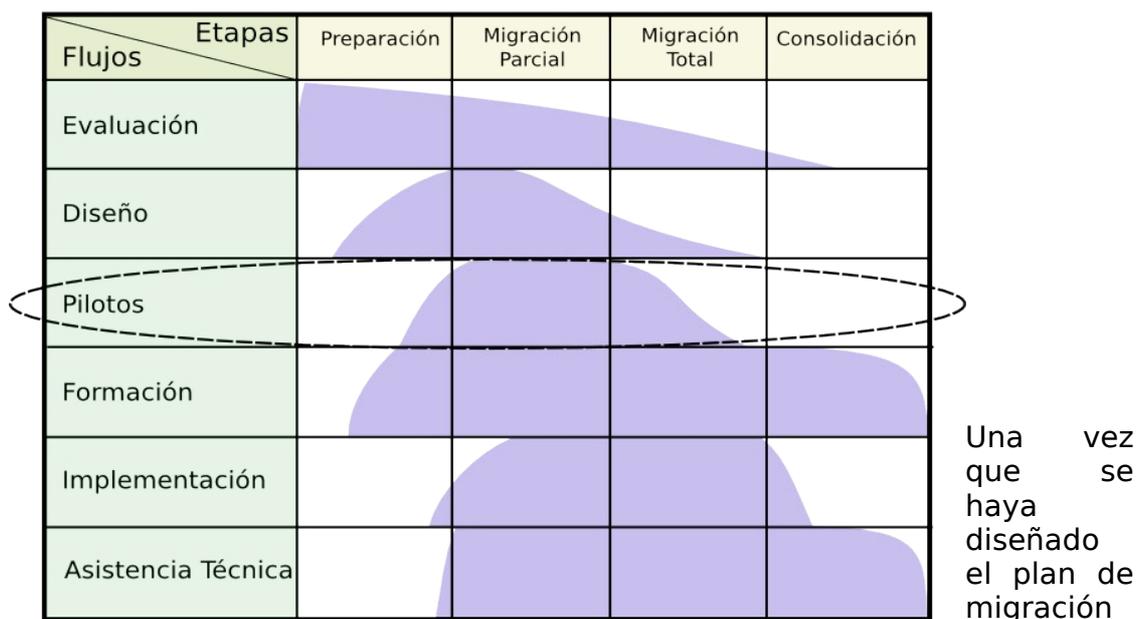
necesidades requeridas como resultados de la evaluación. Dicho flujo tiene su mayor impacto en la etapa de preparación, aunque es importante destacar que el diseño se redefinirá durante la etapa de migración parcial como retroalimentación de dicha fase.

Constituyen tareas importantes a realizar durante este flujo:

- Establecer un plan de acciones concreto que abarque todos los elementos a migrar.
- Definir una estrategia de sensibilización de acuerdo a los distintos grupos de usuarios, haciendo énfasis en aquellos que mayor impacto poseen en la toma de decisiones, buscando lograr una mejor gestión.
- Diseñar una justificación para el proceso basado en las ventajas y desventajas que posee la utilización de Software Libre.
- Definir los números de usuarios que participan en cada fase del plan.
- Establecer los números y sistemas informáticos que migrarán hacia Software Libre.
- Establecer los períodos de implementación y soporte de las aplicaciones.
- Definir el orden de atención a los usuarios durante la formación según el impacto de los mismos.
- Definir la estrategia para el proceso de instalación soporte y distribución de aplicaciones, esto incluye gestionar los recursos humanos necesarios, discos de distribuciones a instalar, repositorios, etc.
- Diseñar la ruta de migración. Implementar cada iteración a pequeña escala y validar la configuración propuesta de las aplicaciones.
- Recoger los elementos importantes para introducir mejoras en los distintos puntos de la guía de migración, para ello pueden utilizarse diversos métodos como la encuesta, conversaciones con los usuarios, revisión de los sistemas en funcionamiento, etc.
- Chequear el plan de acciones propuesto.
- Introducir cambios en la guía de migración propuesta.

Lo más importante de este flujo es, además de lo mencionado anteriormente, que permite desarrollar la experticia y la base de conocimiento necesaria para obtener un modelo replicable de migración eficaz y a corto plazo.

### 4.1.3- Flujo de trabajo: Pilotos.



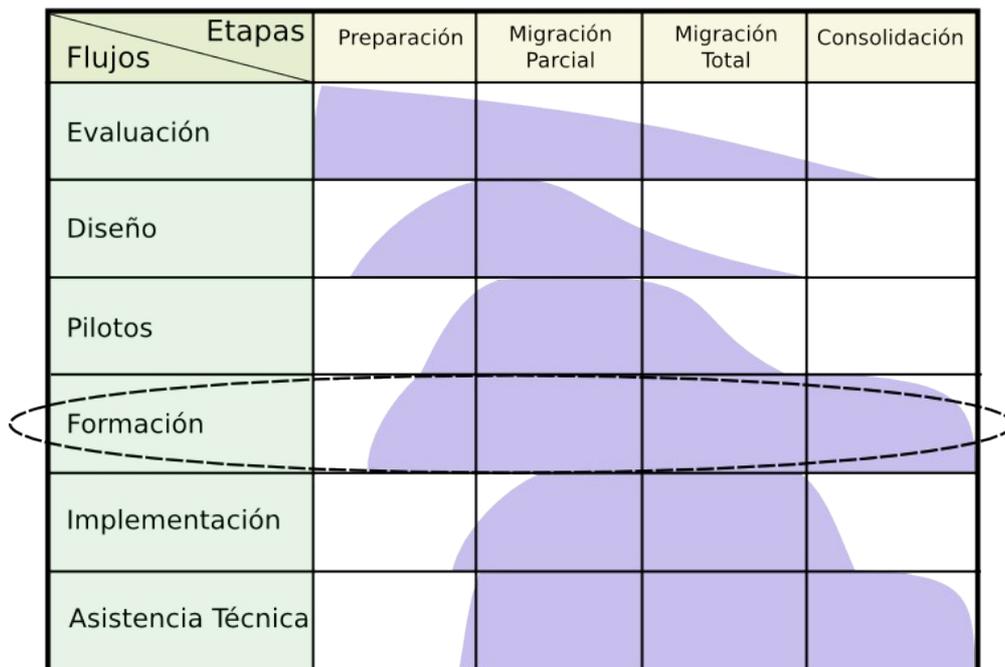
y con el objetivo de validar su contenido y crear las configuraciones correctas para las aplicaciones, es importante crear un ambiente real de pruebas en un marco reducido, que permita la retroalimentación y los ajustes necesarios de algunas variables del plan propuesto, para poder hacer extensiva la migración con la seguridad de que el número de fallos va a ser mínimo. Es en este punto donde se comprueban en tiempo real el plan de migración y el plan de acción institucional.

Las tareas a acometer durante el flujo son:

- Implementar cada iteración a pequeña escala y validar la configuración propuesta de las aplicaciones.
- Recoger los elementos importantes para introducir mejoras en los distintos puntos de la guía de migración, para ello pueden utilizarse diversos métodos como la encuesta, conversaciones con los usuarios, revisión de los sistemas en funcionamiento, etc.
- Chequear el plan de acciones propuesto.
- Introducir cambios en la guía de migración propuesta.

Lo más importante de este flujo es, además de lo mencionado anteriormente, que permite desarrollar la experticia y la base de conocimiento necesaria para obtener un modelo replicable de migración eficaz y a corto plazo.

### 4.1.4- Flujo de trabajo: Formación.



Para lograr que los usuarios acepten la transición, lo más importante es que conozcan el nuevo sistema, por lo que la capacitación se convierte en el baluarte fundamental de la migración. El flujo de trabajo de formación estará presente durante toda la migración, comenzando por la justificación del proceso y abarcando hasta las etapas posteriores al soporte, siendo el objetivo fundamental de la misma la capacitación a todo el personal en los nuevos sistemas informáticos que se implantan en la institución.

Como tareas fundamentales se proponen:

- Confeccionar e impartir planes de formación a los usuarios según el nivel de los mismos, se proponen tres niveles básicos de usuarios que se pueden identificar, esta propuesta puede desglosarse en más niveles en dependencia del lugar:
  - Formación de instructores en Software Libre: El objetivo de estos planes de formación es preparar un grupo de instructores que faciliten el proceso de formación de herramientas libres tanto del personal técnico como de los usuarios finales. Los instructores recibirán cursos prácticos y teóricos de las herramientas que serán utilizadas en los cursos de soporte técnico y usuarios finales.
  - Formación de soporte técnico: Los planes de formación orientados a este contingente tienen la finalidad de preparar un conjunto de personas para dar soporte a los usuarios cuando estos los necesiten a través de diversos métodos, estos usuarios recibirán un plan de formación que contendrá cursos prácticos y teóricos, así como que se les indicarán los mecanismos más adecuados para acceder a los recursos de Software Libre que les garanticen brindar soporte de excelencia.
  - Formación de los usuarios finales: La finalidad de estos planes de formación tienen como objetivo formar en Software Libre a los usuarios finales que usarán las aplicaciones instaladas en el nuevo sistema, para ello se prepararán de igual manera cursos prácticos y teóricos a la medida de las herramientas utilizadas en este grupo pueden incorporarse secretarías, personal administrativos, algún número de estudiantes, entre otros. Se les debe dotar además las principales direcciones de los recursos donde estos puedan apoyarse en dependencia de los medios

establecidos para dar soporte y capacitación.

- Formación de desarrolladores: La formación orientada a los desarrolladores debe contener planes con un grado mayor de conocimientos tanto prácticos como teóricos orientados a este tipo de personas, por lo tanto incluirá cursos de programación en los lenguajes elegidos y sobre el que se desarrollarán las futuras aplicaciones de la institución, se es del criterio de usar como se enuncia en el flujo de asistencia técnica lenguajes y librerías multi-plataformas.

Es importante la creación de un mecanismo de certificación para acreditar cada curso vencido e ir estableciendo determinados niveles en los usuarios para lograr una formación más adecuada.

#### 4.1.5- Flujo de trabajo: Implementación



La implementación es el flujo de trabajo donde se hará efectiva la migración, esta marcará el fin del software privativo según la iteración. De forma paulatina se irán migrando las herramientas e instaurando las aplicaciones y sistemas libres. Es una etapa donde existirán grandes cambios en los sistemas, por lo que la formación y soporte deberá reforzarse. Este será el momento en el que la mayoría de los usuarios podrán tocar de cerca la migración y hacerse partícipes de ella, por lo que todas las medidas que se tomen para el correcto desenvolvimiento de la misma son pocas. Deberán reforzarse las charlas de sensibilización y generar entusiasmo hacia los usuarios con el objetivo de lograr una buena aceptación del proceso.

La implementación de la migración contiene un conjunto de tareas asociadas como otros flujos, algunas de las mismas variarán de un centro a otro en dependencia de las condiciones existentes, las principales tareas concernientes a esta etapa son:

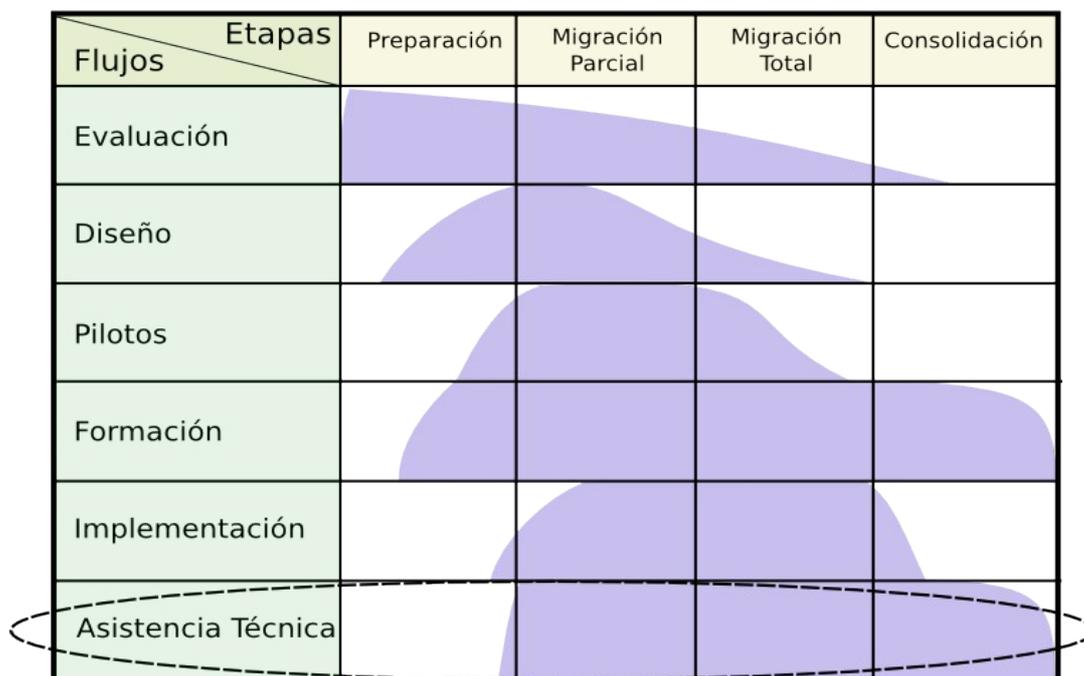
- Hacer efectiva la migración en dependencia de la iteración, introduciendo los cambios respectivos en la tecnología. Cada equipo migrado y configurado

constituye un elemento significativo en el avance de la migración.

- Sustitución de los servidores y dominios de Windows a aplicaciones libres, más adelante, se definen las herramientas libres más utilizadas para el montaje de servidores en Software Libre.
- Instalación de herramientas libres sobre Windows u otro sistema propietario.
- Cambio de Windows o cualquier otro sistema, por la distribución(es) de GNU/Linux que se usará(n) como sistema operativo una vez migrada la institución.
- Fortalecer los planes de formación y certificación de usuarios.
- Fortalecer los mecanismos de soporte y mantenimiento.
- Creación de unidades de desarrollo que puedan servir para el desarrollo de aplicaciones locales y la personalización de herramientas e imágenes de GNU/Linux.
- Movimiento masivo de datos hacia los nuevos formatos y sistemas.

#### 4.1.6- Flujo de trabajo: Asistencia y Soporte Técnico.

El último flujo propuesto en la migración es al igual que la formación uno de los más extensos del proceso. La asistencia y soporte técnico estará presente a todo lo largo del tiempo que dure el cambio. Su objetivo principal es brindar el soporte para el mantenimiento de todas las infraestructuras y al personal. Es vital en toda migración. Su carencia o mal funcionamiento podría provocar la recesión de la migración.



La asistencia y soporte técnico debe estar presente el tiempo que sea necesario de acuerdo a las necesidades del lugar, por lo que el número de iteraciones en la fase de consolidación, donde mayor impacto posee dicho flujo, no está definido a ciencia cierta.

Las tareas y acciones más importantes a llevar a cabo en este momento del proceso son:

- Creación de listas, foros y canales IRC de ser posible para facilitar el debate de temas relacionados.
- Crear canales para la atención a los usuarios vía telefónica ante dudas. Habilitar algún local, stand o espacio físico, a donde puedan dirigirse los usuarios a recibir asesoría.
- Creación y mantenimiento de sitios y cursos virtuales para el auto-aprendizaje y la auto-certificación.
- Creación de repositorios de aplicaciones y datos, y servicios asociados: personalización de repositorios, secciones propias, etc.
- Creación de manuales, FAQs, HowTo.
- Creación, de acuerdo a las posibilidades del centro, de comunidades virtuales en la red.
- Establecer (para los centros en que se realicen desarrollos), políticas que promuevan el uso de estándares para la creación de aplicaciones, como pueden ser librerías gráficas, ejemplo: wxwidgets o foxtoolkit y lenguajes de programación como: ANSI C, Java, PHP y otros.
- Realización de estudios de las aplicaciones más usadas y de las necesidades de los usuarios, a fin de lograr mejoras en el servicio de aplicaciones, optimización de las mismas para aumentar su rendimiento y saber en cada momento cuales son las necesidades de los clientes.
- Creación de un portal informativo como centro del proceso que integre los métodos mencionados anteriormente y contenga un Service-Desk o escritorio de servicios, para aumentar los tiempos de respuesta ante las preguntas más frecuentes.

Lograr la automatización de la mayor cantidad de tareas, por ejemplo implantar mecanismos para la actualización automática de los repositorios, instalar herramientas que permitan la instalación automática de herramientas en las maquinas de los clientes, etc.

#### **4.2- Alternativas de migración.**

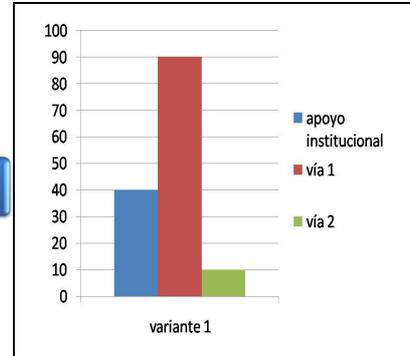
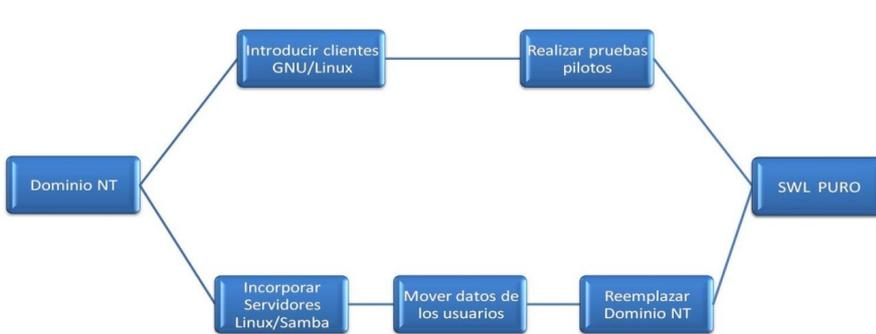
En la determinación de las alternativas de migración de los sistemas de información a software libre, será necesario analizar una serie de elementos, entre los que se encuentran:

- La disponibilidad presupuestal con la que cuenta la institución para la ejecución del proyecto de migración.
- La factibilidad total o parcial de migración de los sistemas de información existentes en la institución.
- El hardware que contiene a los sistemas de información.
- La cantidad y calidad de los recursos humanos disponibles.

La bibliografía consultada coincide en la existencia de 2 rutas para la realización de la migración. No resulta completamente acertada la propuesta que hace el estudio europeo, en la que se insta a utilizar la 2da ruta de migración, alegando que es mucho más sencilla que ejecutar un sistema mixto GNU/LINUX /Windows. Existe un factor que no debe olvidarse y es el apoyo que brinda la institución al proceso, tanto en presupuesto como en gestión. Si no es completo o simplemente no se cuenta con este, entonces esa vía no es conveniente y obliga a decidirse por la 1ra ruta, cuya práctica ha demostrado, no ser del todo difícil.

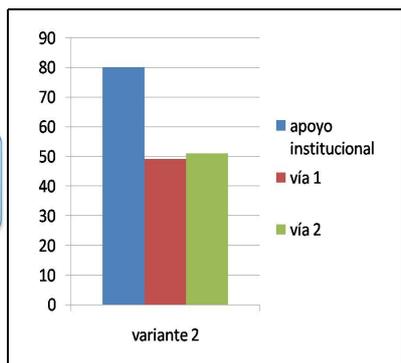
## RUTA 1

Añadir estaciones GNU/LINUX a los dominios Windows existentes e ir trasladando gradualmente los datos y los usuarios, y luego eliminar 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 sistema libre.



## RUTA 2

Construir una infraestructura paralela de tipo GNU/LINUX 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 GNU/LINUX-Windows, pero crea una cooperación entre la gente que usa Windows y la que usa sistemas GNU/LINUX más difícil. (31)



# **Parte 5**

## **ETAPA DE**

### **PREPARACIÓN.**

## 5- ETAPA DE PREPARACIÓN.

Se realizará en esta, la planificación global, donde es necesario que tanto la alta dirección de las instituciones, las áreas de gestión informática y los usuarios responsables de los sistemas estratégicos de la institución, tomen conciencia de la importancia de acometer la migración. En un primer momento se trabajará en la elaboración de un Plan de Migración que deberá ser respaldado por el Plan de Acción Institucional, compromiso de apoyo de la alta dirección con la implementación de los procesos.

El Plan de migración está conformado por una serie de acciones agrupadas estratégicamente en etapas o fases, para lograr migrar los sistemas operativos, aplicaciones y herramientas informáticas que poseen las entidades públicas a Software Libre, con la participación activa del área informática y diferentes dependencias usuarias encargadas de generar, procesar, mantener y aplicar la información.

Algo que resulta imprescindible y que propone esta guía para dar inicio al proceso de migración a Software Libre, consiste en efectuar dos tareas importantes: la sensibilización institucional respecto a la migración a Software Libre y la organización institucional para la implementación del Software Libre. Todos estos procesos tienen como denominador común la necesidad de realizar previamente la capacitación y diagnóstico de los sistemas de información de las entidades del estado, a fin de registrar todos los sistemas de información que posee la institución, y evaluar el grado de factibilidad para la migración a Software Libre.

Se chequearán:

- las herramientas o aplicaciones y sus equivalente en GNU/Linux,
- el grado de seguridad,
- la confiabilidad de la información,
- la información técnica disponible,
- el soporte que ofrecen las distribuciones más conocidas y recomendadas,
- los recursos humanos de que dispone la institución para el proceso.

Lo anterior será logrado a través de un levantamiento o inventario informático que incluya:

- Inventario de los equipos informáticos de la institución.
- Inventario del software informático disponible en la institución.
- Inventario de los equipos empotrados.
- Inventario de los sistemas de información o aplicaciones utilizados por la institución.

Y no debe faltar en esta etapa un inventario del conocimiento especializado del personal de informática existente en la institución, que serán los encargados de la implementación del plan de migración de los sistemas de información a Software Libre.

En esta etapa se elaboraran el Plan de acción y los Cronogramas de ejecución, los que comprenderán una estimación final de los costos implicados, y se insertarán en el Plan de Acción Institucional. Se incluyen además la asignación de recursos humanos, soporte tecnológico y partida presupuestaria lo que será controlado por la alta dirección de la institución, que debe velar por ello para que se cumplan las actividades y cronogramas establecidos. (32)

Cierra esta etapa con la valoración del costo de la migración.

### 5.1- Levantamiento de Información.

Para la realización del Levantamiento de Información se pueden emplear todo tipo de variantes incluido el uso de planillas impresas que recojan los principales datos de interés. Lo importante es que se disponga de toda la información posible y pueda ser procesada por el equipo técnico en la toma de decisiones.

INVENTARIO DE LAS APLICACIONES INSTALADAS EN LAS COMPUTADORAS DE LA INSTITUCIÓN												
INSTITUCIÓN:						FECHA:						
DIRECCIÓN U OFICINA:						INFORMACIÓN PROPORCIONADA POR:						
Nombre de las aplicaciones instaladas	Sistema operativo Base	Lenguaje de Programación	Versión	Orientación de la aplicación	Dispone de Documentación (S/N)	Dispone los programas fuentes (S/N)	Fabricante					
INVENTARIO DE SOFTWARE DE BASES DE DATOS INSTALADOS EN LA INSTITUCIÓN												
INSTITUCIÓN:						FECHA:						
DIRECCIÓN U OFICINA:						INFORMACIÓN PROPORCIONADA POR:						
Nombre del software de Base de Datos	Sistema operativo Base	Tipo de archivo (extensión)	Versión	Cantidad de registros en el archivo/ BD	Tamaño del archivo (Kb)	Sistema(s) que lo utilizan	Fabricante					
INVENTARIO DE HARDWARE DE LAS COMPUTADORAS DE LA INSTITUCIÓN												
INSTITUCIÓN:						FECHA:						
DIRECCIÓN U OFICINA:						INFORMACIÓN PROPORCIONADA POR:						
Microprocesador		Mainboard	Memoria RAM (MB)	Discos duros				Tarjetas de video		Tipo de Mouse	Tipo de teclado	Tipo de tarjeta de red
Tipo	velocidad			Número	Tipo IDE o SCSI	Cantidad de C/U (GB)	Número de particiones	Tipo	Memoria			
INVENTARIO DE SOFTWARES INSTALADOS EN LAS COMPUTADORAS DE LA INSTITUCIÓN												
INSTITUCIÓN:						FECHA:						
DIRECCIÓN U OFICINA:						INFORMACIÓN PROPORCIONADA POR:						
Nombre del software instalados	Sistema operativo Base	Fabricante	Versión	Orientación del software	Modos de instalación	Extension(es) de trabajo predeterminado	Número de licencias					

Esta aplicación y servidor

son una aplicación y servidor que se ejecuta a

transformaciones, a fin de generar los reportes de información necesarios.

**Name:** RAMON08  
**Domain:** INICIOMS  
**Userdomain:** RAMON08  
**Last inventory:** 05/28/2008 21:11:01  
**IP address:** 100.100.100.34  
**User:** RamÃ³n Paumier SamÃ¡n  
**Memory:** 1024  
**Swap:** 2167  
**Network name 1:** 0.0.0.0  
**Network name 2:** 100.0.0.0

**OS Name:** Microsoft Windows XP Professional  
**OS Version:** 5.1.2600  
**Service pack:** Service Pack 2  
**Comments:** COPEXTEL  
**Windows user:** MPC  
**Windows licence:** 55274-642-4774706-23939  
**Windows key:** 3XDGP-YVTMX-MMM2D-M6FJB-JD33V  
**User agent:** OCS-NG\_windows\_client\_v4046

**SOFTWARE**

Editor	Name	Version	Comments
Adobe Systems, Inc.	Adobe Acrobat 5.0	5.0	N/A
Adobe Systems	Adobe Acrobat 8 Professional - English, FranÃ§ais, Deutsch	8.0.0	N/A
Adobe Systems Incorporated	Adobe Flash Player ActiveX	9.0.115.0	N/A
Adobe Systems Incorporated	Adobe Flash Player Plugin	9.0.115.0	N/A
	ATI - Utilidad de desinstalaciÃ³n de software	6.14.10.1014	N/A
1st Benison	All Converter 5.0.6		N/A
	ATI Display Driver	8.283-060808a-036121C-Asus	N/A
	AVG 7.5		N/A
boilsoft, Inc.	AVI/MPEG/ASF/WMV Splitter 2.31		N/A
Microsoft			

**PROCESSOR(S)**

Type	Processor Speed (Mhz)	Number
Intel(R) Core(TM) Duo CPU T2350 @ 1.86GHz	1866	2

**MEMORY**

Caption	Description	Capacity (MB)	Purpose	Type	Speed	Slot number
Matriz de memoria fÃ¡sica	Matriz de memoria fÃ¡sica	0	System Memory	Empty slot		2
Memoria fÃ¡sica	DIMM1 (No ECC)	1024	System Memory	Unknown		1

**STORAGE**

Name	Manufacturer	Model	Description	Type	Disk size (MB)
FUJITSU MHV2120AH	(Unidades de disco estÃ¡ndar)	\\.\PHYSICALDRIVE0	Unidad de disco	Fixedhard disk media	114470
TSSTcorp CD/DVDW TS-L632D	(Unidades de CD-ROM estÃ¡ndar)	TSSTcorp CD/DVDW TS-L632D	Unidad de CD-ROM	CD-ROM	0

**DISK(S)**

Letter	Type	File System	Total (MB)	Free (MB)	Designation
C:/	Hard Drive	NTFS	20104	11164	
D:/	CD-Rom Drive		0	0	
F:/	Hard Drive	NTFS	54093	13717	DATOS

**BIOS**

5.  
Es

y  
y

- Informe de la situación de la entidad (a partir del Inventario) para la realización de la migración.
- La forma en la que se migrarán los servicios a partir de las particularidades de la entidad.
- La forma en la que se migrarán las bases de datos a partir de las particularidades de la entidad.
- La forma en la que se migrarán las estaciones de trabajo a partir de las particularidades de la entidad.
- Aplicaciones informáticas que sustituirán a las utilizadas en software privativo, así como, la forma en que se emularán las que no tengan equivalente libre y los pasos a acometer para la migración de los sistemas de gestión adquiridos a empresas productoras o producidos por personal del propio centro.
- Planificación de tiempo, recursos materiales, personas involucradas.
- Plan de capacitación del personal: estrategias de capacitación, materias de capacitación.
- Forma en que se dará soporte.
- Estimaciones Generales: costos, tiempo, número de personas, recursos.

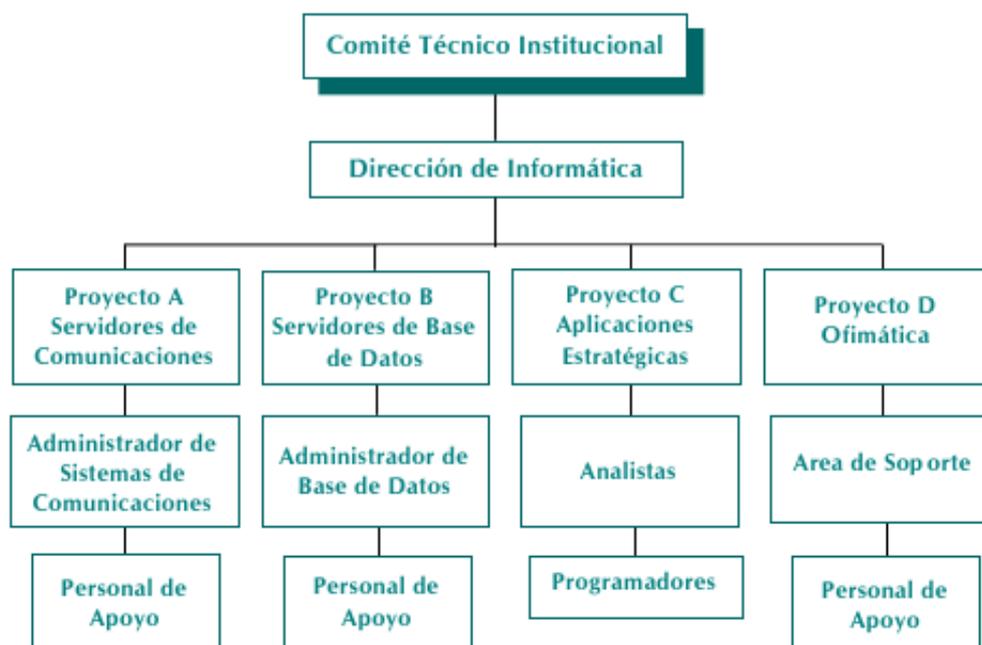
### **5.3- Plan de acción institucional.**

Se elabora por parte del área de informática de la institución a partir del análisis del Plan de migración elaborado previamente por los especialistas informáticos. Su objetivo esencial es lograr la concientización y el comprometimiento de los directivos y trabajadores de la entidad con el proceso de migración a Software Libre: lo que se traduce en asegurar el cumplimiento de lo planificado en el plan de migración a partir del aseguramiento de los recursos, la designación de responsables de cada tarea y la determinación del fondo de tiempo para el cumplimiento de cada una.

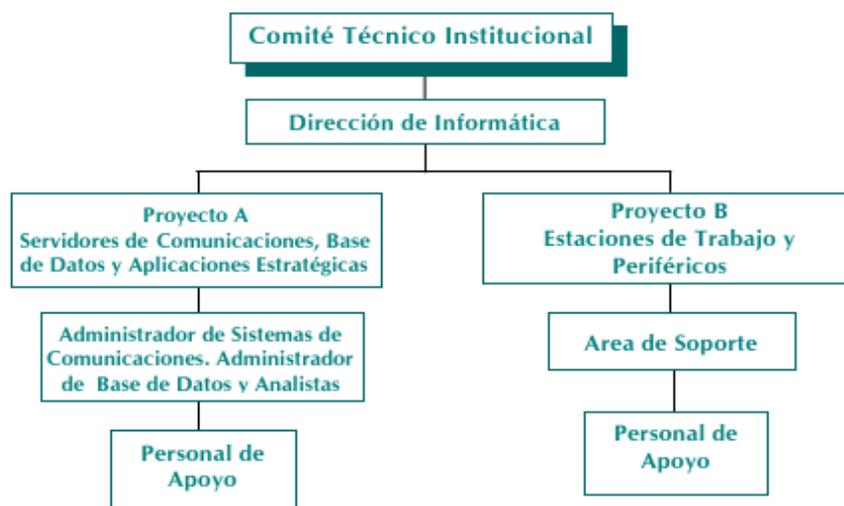
Este documento, junto al Plan de migración, serán sometidos a una prueba piloto en la próxima etapa de Migración Parcial, a fin de comprobar en tiempo real y a pequeña escala, la correcta elaboración del primero y la efectividad y cumplimiento del segundo.

El plan de acción aprobado deberá recoger las acciones, tareas y funciones que correspondan a cada estructura orgánica de la institución. Las funciones del Área de informática estarán orientadas a la ejecución del aspecto técnico, dentro de los plazos requeridos y enmarcados dentro de la partida presupuestaria destinada al proyecto.

#### **5.3.1 Estructura organizativa de una gran Entidad para la Migración.**



### 5.3.2 Estructura organizativa de una pequeña Entidad para la Migración.



### 5.4- Laboratorio de formación, capacitación y soporte.

Las labores relacionadas con la formación y capacitación del personal, tienen sus primeras acciones en esta etapa, una vez que se ha procesado el potencial de recursos humanos con que se cuenta.

Se indica para esta etapa, la clasificación de todo el personal, a fin de enfocar su capacitación individual, a uno de los perfiles descritos anteriormente en la explicación del flujo de trabajo y cuyos cursos particulares para cada tipo de usuario se incluyen en el metapaquete que acompaña a esta guía.

En la medida de lo posible, cada institución debería ser capaz de montar un laboratorio con un determinado número de computadoras con el objetivo de convertirlo en cuartel general de todo el proceso, zona de las pruebas piloto, centro de la capacitación y lugar de soporte.

# **Parte 6**

# **ETAPA DE MIGRACIÓN**

# **PARCIAL.**

## 6- ETAPA DE MIGRACIÓN PARCIAL.

La metodología propone para las etapas de Migración Parcial y Migración Total tres iteraciones en cada una de ellas. La primera está orientada a la implantación de Software Libre en los servidores de la institución. Es importante esto como primer paso, pues de esta forma se contribuye a disminuir una de las grandes barreras de proceso de migración “el miedo a perder el poder”, enunciado en las Directrices IDA de la UE.

Luego de tener la infraestructura de servidores completamente en FLOSS (Free Libre Open Source Software), como próxima iteración se propone el uso de herramientas libres en el sistema actual (Microsoft Windows), esto proporcionará que los usuarios puedan probar las herramientas con las que trabajarán en GNU/Linux en un entorno ya conocido, teniendo la posibilidad de capacitarlos en estas herramientas sobre este entorno y de ir facilitando la conversión de archivos a formatos estándares, para evitar problemas de compatibilidad entre las aplicaciones. Existe una lista bastante amplia de aplicaciones libres para Windows ( [www.cdlibre.org](http://www.cdlibre.org)) que pueden sustituir a todas las aplicaciones que comúnmente se utilizan. (33)

Una vez que estén los usuarios ya preparados en las herramientas libres que usarán sobre GNU/Linux y convencidos de las ventajas de utilizar FLOSS se propone el cambio de la base y la instalación de GNU/Linux como nuevo sistema operativo a usar, lo que supone gran atención por todos los implicados en la tarea, pues es aquí donde más “frágil” resulta la migración.

Complementariamente a las dos primeras iteraciones, se puede utilizar la virtualización para algunas herramientas, con el objetivo de hacer pruebas. Es importante no dejar escapar que la migración tal vez no se realice de forma total, debido a la existencia de aplicaciones que no poseen equivalente libre, en tal caso se pueden utilizar aplicaciones como Wine, que permiten emular software de Windows sobre GNU/Linux; unificar varias herramientas libres para lograr los mismos resultados que la propietaria o la re-programación de la misma.

### **6.1- Migración Parcial 1 de Servidores y Bases de Datos.**

En nuestros días las soluciones tecnológicas basadas en software libre han crecido lo suficiente como para convertirse en una alternativa sólida que compite con carácter de igual con otras tecnologías de software propietario como Microsoft. Es probable que la batalla con preferencia de los clientes de servidores para empresas esté en su mejor momento. Los Servidores GNU/LINUX pueden, a priori, asustar a algunos administradores; no obstante, GNU/LINUX provee de paneles de control, para la gestión, configuración y administración de los servidores. De esta manera, sin conocimientos avanzados de este sistema, se puede gestionar, administrar y configurar un servidor de forma sencilla e intuitiva.

Los servidores GNU/LINUX serán siempre más económicos que los servidores Microsoft Windows en aspectos como la actualización, desarrollo, soporte, mantenimiento del software y la prolongación de vida útil del hardware. La principal razón de ello es que existe una gran comunidad de software libre y aplicaciones gratuitas, o sea que se ahorra en lo referente al pago de las licencias. Tanto el sistema operativo GNU/LINUX como las aplicaciones que normalmente corren sobre él, no requieren de licencias del proveedor.

Según una investigación del banco de negocios The Goldman Sachs Group sobre 100 de las mayores empresas de los EEUU, GNU/LINUX está ganando fuerza en casi todos los terrenos: servidores de impresión, correo electrónico, Web y equipos de sobremesa. Incluso en aplicaciones mucho más críticas, como en mainframes, bases de datos y servidores de aplicaciones, lo que hace que este sistema operativo de coste 0, gran fiabilidad en cuanto a velocidad de procesamiento y seguridad gane cada día más auge y sea en la actualidad el sistema operativo ideal para asegurar cualquier tipo de datos.

Los servidores más usados a nivel mundial albergan sistemas operativos libres basados en una distribución de GNU/LINUX. Por supuesto, la interrogante enseguida nos asalta. Lo primero que podemos alegar es que no es necesario pagar por su uso ni por la mayoría del software disponible para esta plataforma o sistema operativo. Son mucho menos afectados por virus, troyanos, gusanos lo que impacta sensiblemente en los costos de mantenimiento y en la calidad del servicio ofrecido; además existen potentes antivirus tan eficientes como los que existen en los servidores de software propietario.

Aprovecha mejor las características del hardware, pudiendo ejecutarse aún con configuraciones no aptas para el uso de otros sistemas. Se incluyen funciones que en otros sistemas se necesita de la instalación de software adicional. Su administración se puede automatizar, con lo cual se obtiene una mayor productividad del personal a cargo del mantenimiento. Debido a todo lo expuesto anteriormente grandes empresas a nivel mundial lo usan entre las cuales podemos citar:

- Google: El principal proveedor de servicios en Internet lo utiliza en todos sus servidores y en sus puestos de trabajo.
- IBM: Lo utiliza en servidores, sistemas de fabricación y puestos de trabajo.
- HP: Provee soporte para sus productos y mantiene desarrolladores del proyecto Debian.
- Dreamworks, Pixar, Industrial Light and Magic y Digital Domain: Lo utilizan en la producción de efectos especiales para cine.
- Sony: Colabora en su desarrollo para dispositivos domésticos.
- Wall Street: Merrill Lynch y Credite Suisse First Boston: lo utilizan para sus transacciones.
- SAP y Oracle: Lo utilizan en sus servidores y en el desarrollo de sus aplicaciones.
- Alemania, China, Brasil, España, Israel, Argentina: Lo utilizan en distintos ámbitos de la administración pública.
- NASA: Lo utiliza, inclusive en misiones críticas. [34]

### **6.1.1- Migración Parcial de Servidores.**

La migración de los servidores propietarios hacia los servidores con sistema operativo GNU/Linux se hará gradualmente. Los servidores propietarios dejarán de funcionar una vez que el servidor con el sistema operativo GNU/Linux se encuentre instalado y correctamente migrado. Estas acciones serán completamente transparentes a los usuarios de la red, pues la misma funcionará con los mismos servicios de igual forma. La administración de estos servidores podría realizarse a través de los propios ficheros de configuración o a través de aplicaciones con interfaz gráfica tales como *webmin* que facilitan esta labor.

### 6.1.1.1- Propuestas de migración de un servidor DNS

Al hablar sobre la migración hacia un sistema de nombres de dominio o servidor de nombres y dominios como lo indican sus siglas en inglés de Domain Name Service debemos pensar en el BIND, el conocido Berkeley Internet Name Domain. Este el servidor de DNS más popular de Internet, es el más utilizado por su calidad y su característica de ser multiplataforma.

BIND no puede decirse que sea complicado sino que es un producto que funciona a través de una serie de archivos de configuración y que aún sin mucha práctica, está al alcance de cualquiera, además, hay grandes volúmenes de documentación sobre este producto, lo cual lo hace ser muy conocido y usado. El mismo cuenta con todas las características necesarias para ejecutar este tipo de software de servidor, la cuales son un espacio de nombres jerárquico para los hosts y las direcciones IP, un "resolvedor" o librería de rutinas que permite realizar consultas a esa base de datos, enrutamiento mejorado para el correo electrónico, un mecanismo para encontrar los servicios en una red, un protocolo para intercambiar información de nombres. El mismo brinda los servicios de resolución de nombres a direcciones IP y resolución inversa (de direcciones IP a nombres), así como listas de control de acceso, localización de servicios (registros SRV – RFC2052-), respuestas parametrizadas en función del origen de la petición conocidas como vistas, así como logs que nos brindan diferentes informaciones acerca del funcionamiento del servidor y las peticiones al mismo.

Este servidor de nombre de domino creado al comienzo de la Internet ya se encuentra en su versión 9, una versión muy estable y que gracias a la libertad de su código ha avanzado vertiginosamente librándose de innumerables vulnerabilidades constituyendo actualmente una de las mejores base de datos distribuida y jerárquica que almacena información relativa a los nombres de dominio en internet o gestiona nombres de equipos y servicios en redes locales; podemos citar como ejemplo de la gran magnitud usabilidad a nivel internacional de este servidor las pruebas realizadas por el sitio <http://dns.measurement-factory.com/> especializado en tareas de recaudación de estadísticas en Octubre del 2007 donde fueron comprobados 386 588 dominios de internet , de los cuales el 64% usaba la versión 9 de BND para "resolver" sus nombres de dominio, otras estadísticas sobre el uso del mismo y comparativas entre uso de servidores DNS pueden ser comprobadas en la web de netcraft. [35]

Existen diferentes vías para migrar un servidor DNS desde Windows hacia GNU/LINUX. Si existe un número pequeño de entradas del DNS, la migración se puede efectuar copiando y pegando la información, ya que esta puede obtenerse en el Microsoft DNS Manager o en el archivo (.dns) que se encuentra en:

```
C:\WINNT\system32\dns\
```

Sin embargo este método puede consumir mucho tiempo si se tratase de muchas entradas. Para muchos la forma más fácil de transferir la información es usando el mecanismo de Transferencia de Zona, bastante útil y efectiva. Otra de las formas es haciendo uso de algunos script que interpretan la configuración en el servidor de Windows y generan la configuración para BIND, uno de ellos es:

## w2lmt-migrate-dns

El cual se puede descargar a través de la web y pertenece a un conjunto de scripts realizados a raíz de la publicación del libro “Windows to GNU/Linux migration toolkit” de David Allen, los mismos se encuentran disponibles a través de la URL que se mostrará al final de la explicación. Este método automatiza el proceso, lo hace más rápido y factible cuando tenemos un gran número de entradas. Para el uso de este script lo primero a tener en cuenta será configurar el archivo llamado *migrate-dns.conf*, con la configuración de este archivo vamos a obtener las bases de datos del servidor de dominio, las cuales serán almacenadas en una dirección especificada. Posteriormente ejecutamos el comando:

## w2lmt-migrate-dns -f migrate-dns.conf

Con esto ya tenemos una copia de nuestro DNS. Debemos tener en cuenta que en el servidor al cual queremos hacer una réplica de sus registros debemos permitir a través de la herramienta administrativa usada para el mismo transferir sus zonas hacia el host especificado. Después y una vez ejecutados los scripts de migración debemos pasar a configurar nuestro DNS previamente instalado con los datos adquiridos. Teniendo en cuenta que es muy posible que tengamos que modificar los registros SOA y NS producto del nuevo cambio de servidor (dirección ip).

### 6.1.1.2- Propuestas de migración hacia un servidor ldap.

La propuesta de migración de un Active Directory hacia un servidor libre se basa en la implementación libre del protocolo de acceso ligero a directorios más conocida como OpenLDAP. El mismo es un servicio de directorio que, entre otras cosas, nos permite contener los datos (logins, claves) de una serie de usuarios y realizar la autenticación en máquinas clientes a través de un único servidor OpenLDAP. Algunas características del mismo por las cuales se propone son:

- Es muy rápido en la lectura de registros
- Permite replicar el servidor de forma muy sencilla y económica
- Muchas aplicaciones de todo tipo tienen interfaces de conexión hacia OpenLDAP y se pueden integrar fácilmente
- Dispone de un modelo de nombres globales que asegura que todas las entradas son únicas.
- Usa un sistema jerárquico de almacenamiento de información.
- Permite múltiples directorios independientes.
- Funciona sobre TCP/IP y SSL.
- Es un servidor fácil de instalar, mantener y optimizar.
- Proporciona respuestas rápidas a operaciones de búsqueda o consulta.
- Tiene la capacidad de replicar información, con lo que aumenta la disponibilidad y fiabilidad de la información, reduciendo además el tiempo de respuesta.
- Presenta varios clientes que permiten una interfaz gráfica amigable del mismo.
- Combinado con Samba, puede ser usado como controlador de dominio cumpliendo muchas de las exigencias del Active Directory.

Acerca de las razones expuestas anteriormente podemos decir de este protocolo para mantener e intercambiar información almacenada en directorios con grandes

funcionalidades y provisto de una implementación libre, que es en muchísimos aspectos una herramienta potente que cumple su función, brindando grandes oportunidades a quienes deseen realizar la migración de cualquier organización, contando además con elevados niveles de eficiencia, seguridad y posibilidades de automatización.

Para lograr la migración desde un Servicio de directorios activo de Windows Server 2003 hacia un OpenLDAP de GNU/Linux, se hace necesario primeramente lograr la instalación de un Controlador de Dominio Primario con samba, para permitir el acceso al mismo por cualquier tipo de plataforma que usen los clientes, ya sea Windows o GNU/LINUX.

Como realizar la migración:

Como se explica anteriormente, lo primero es crear el controlador de dominio, esto se hará a través de scripts implementados en perl, los cuales han sido implementados por el equipo de desarrollo del libro de David Allen "Windows to Linux Migration toolokit" que permitirán conectarse al controlador de dominio primario (PDC) sobre Windows y hacer un respaldo del mismo a través de samba que debe estar configurada como BDC. Los pasos a seguir para lograr este objetivo son:

1. Instalar y configurar adecuadamente el servidor Openldap teniendo en cuenta que crearemos el controlador de dominios a través de samba.
2. Configurar el mecanismo de autenticación de la máquina que está siendo usada como servidor, para de esta forma usar el servicio de autenticación de los usuarios a través del protocolo LDAP.
3. Instalar y configurar samba teniendo en cuenta que será usado como controlador de dominio secundario (BDC), o controlador de dominio que será usado para realizar una copia del AD sobre Windows Server.
4. Instalar smbldaptools para proveer de esta forma las herramientas necesarias para lograr establecer la comunicación entre Samba y Openldap. No es necesario la configuración de las mismas pues este proceso se automatizará.
5. Configurar los scripts de acuerdo a las características del AD, es decir, el ip del mismo, la base de búsqueda y demás.

Ya debemos estar listos para comenzar el proceso de migración de la forma más segura posible. Para ello debemos asegurarnos que Samba no esté ejecutándose y sí lo esté Openldap y proceder a ejecutarlos.

Llegado este momento, ya se debe estar listo para convertir al servidor Samba en controlador de dominio. Para ello es necesario activarlo por uno mismo en el servidor y reiniciarlo.

Una vez migrado el directorio activo, es posible que algunos servicios que autentican contra el antiguo Active Directory, y sea necesarios modificarles algunas configuraciones para que funcionen correctamente, lo cual seria conveniente evaluar antes de la migración y cambiar las configuraciones de estos servicios simultáneamente. También vale aclarar que en el momento de producirse la migración del AD, estos servicios que autenticaron con el antiguo AD, y que aun continúan sobre Windows Server es necesario que cuanto antes sean reemplazados por soluciones libres sobre GNU/Linux, tal es el caso de las aplicaciones web, servicios de mensajería instantánea, servidores ftp, de correo y proxy.

### **6.1.1.3- Propuesta para migrar el DHCP .**

DHCP es un protocolo 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 IP y reasignará las direcciones siempre que sea posible. También permite la administración central de muchas direcciones globales como servidores de portales y nombres.

La alternativa libre propuesta para este servidor es el dhcp3-server.

El primer paso para migrar DHCP desde Windows hacia GNU/LINUX consiste en determinar todos los ámbitos así como las propiedades de cada uno de ellos, esto lo hacemos a través de la interfaz de administración del servidor DHCP, donde debemos encontrar los ámbitos que tenemos en nuestro servidor y obtener la información de cada uno de ellos. Esta información la guardamos en un fichero o bien la podemos anotar en una hoja de papel. Después de recolectar toda la información de los ámbitos entonces podemos proceder a configurar el DHCP en GNU/LINUX. Este proceso puede automatizarse, ahorrándonos mucho más tiempo y consiste en ir al servidor y exportar la lista de ámbitos desde la herramienta que permite administrar el servidor de dhcp; con esta lista de ámbitos, algo muy sencillo y viable es hacer un pequeño script en perl u otro lenguaje, que con esta información nos coloque todos los parámetros necesarios para que nuestro nuevo servidor de DHCP sobre GNU/Linux lo use. La parte de la configuración consiste en editar el fichero dhcpd.conf con la información de los ámbitos, información que introduciremos a través del script anteriormente mencionado. Por último reiniciamos el servicio y comprobamos que el DHCP que acabamos de montar sobre GNU/LINUX es funcional, pruebas realizadas con este sistema han sido satisfactorias en grandes redes de computadoras.

### **6.1.1.4- Propuesta para servicios de impresión.**

En GNU/LINUX el servicio de impresión se logra a través de samba, el cual es usado para poder compartir las impresoras entre toda la red y CUPS para administrirlas. Samba como conocemos es una implementación libre del protocolo de archivos compartidos de Microsoft Windows para sistemas de tipo UNIX, y es capaz de compartir directorios, autenticando con su propio archivo de usuarios así como servir colas de impresión, para lo cual hemos hecho alusión acerca de samba.

CUPS está compuesto por una cola de impresión con su planificador, un sistema de filtros que convierte datos para imprimir hacia formatos que la impresora conozca, y un sistema de soporte que envía los datos al dispositivo de impresión. CUPS utiliza el protocolo IPP (Internet Printing Protocol) como base para el manejo de tareas de impresión y de colas de impresión. Posee un amplio soporte sobre operaciones bajo el protocolo SMB. Existen varias interfaces de usuario para diferentes plataformas para configurar CUPS, la misma cuenta también con una interfaz como aplicación Web. CUPS se distribuye bajo licencia GNU General Public License.

Instalar y configurar el servicio de impresión usando samba y CUPS para administrar las impresoras cuenta de varios pasos, los cuales se mencionan a continuación:

-Conocer como se realiza la Impresión remota en Windows:

Cuando un programa quiere imprimir a una impresora accesible a través de SMB, el sistema se comporta exactamente igual que en el caso de una impresión local, excepto que spool32 utiliza la funcionalidad de la red para enviar los datos listos para imprimir al spool32 de la máquina donde está físicamente enchufada la impresora. El

pool32 de la máquina remota es el que enviará los datos al puerto. Si el servidor de impresión resulta ser Samba y no Windows, entonces Samba invocará lpr y le inyectará transparentemente lo que Windows esté enviando. Luego lpr hará su trabajo en la forma habitual. El trabajo de impresión ya llega a Samba en el formato que la impresora requiere, así que el demonio de impresión de Unix no debe pasar tales datos por ningún filtro

-Conocer como se realiza la impresión remota en GNU/LINUX:

El archivo printcap puede definir una cola de impresión remota, cuando lpd va a enviar los datos a la impresora, en lugar de mandarlos al dispositivo donde estará conectada la impresora, los manda a un puerto TCP de una máquina remota donde se supone que otro lpd está escuchando. Esto es bastante más simple y efectivo que pasar los datos por Samba, se puede controlar que hosts y usuarios pueden imprimir usando la autenticación nativa de Unix, y demás.

-Definir las colas locales de impresión:

En este paso como su nombre lo indica se definen las colas de impresión, lo cual se logra a través de los ficheros de configuración que nos brindan los programas que las manipulan.

-Hacer accesible la cola a través de la red:

Este paso cuenta de dos etapas, una sería hacer accesible la cola de impresión desde GNU/LINUX a través de un comando, para personalizarlas se le deben dar permisos a las distintas partes de CUPS a través de la modificación de un fichero. La otra etapa sería hacerla accesible desde Windows, así que modificaremos un archivo, el cual contendrá los nombres de las colas de impresión, cualquier otra información proporcionada es ignorada por samba

-Etapa Final

Configurar la impresora y la máquina cliente para que esta pueda imprimir en remoto, teniendo en cuenta que este tipo de servicio se realiza indistintamente dependiendo del sistema operativo que se use, incluso de la distribución de GNU/Linux.

#### **6.1.1.5- Propuestas de migración de un servidor de correo.**

Existen varias alternativas libres, alguna de las cuales son: Sendmail, Qmail, Postfix y Exim. Se propone el uso de Postfix como servidor de correo. Principalmente debido a que su configuración es sencilla y bastante flexible; es mucho más estable, seguro y de capacidad ilimitada de cuentas; posee gran facilidad de configuración, no siendo así en Sendmail que es más compleja la configuración. Las aplicaciones para correo electrónico como pueden ser los clientes de correo, se integran fácilmente con Postfix.

Se le pueden incluir listas de correo, antivirus, anti spam, interface webmail y otras características. Es un servidor de correo modular y cada módulo tiene una tarea diferente con un mínimo intercambio de información entre ellos, lo que aumenta su seguridad. Además posee tablas de acceso rápido en múltiples formatos y tiene una gran integración con Ldap, Mysql, PostgreSQL y diversas bases de datos relacionales. Postfix posee soporte de TLS (Transport Layer Security), mbox, maildir, dominios virtuales, SASL y reescritura de dirección; y una gran capacidad para manejar altos volúmenes de correo.

Una buena combinación para obtener un servidor de correo es Popsfix como servidor de correo SMTP, Cyrus-SASL como servidor POP/IMAP con SSL, squirrelmail como servidor de Webmail, Clamav+Amavis-new como sistema de antivirus, y Postfix VDA para dar cuotas de disco a los buzones.

En los lugares donde aun exista el servidor correos Microsoft Exchange y por políticas de la entidad haya que mantener dicho servicio, se puede usar la alternativa de código abierto Open-Xchange, que es la solución de trabajo en grupo open source más utilizada en todo el mundo. Con la migración a este servidor se podrán mantener los datos de los usuarios y el cambio será transparente para ellos, podrán continuar usando sus clientes de correo como Outlook o Thunderbird y disfrutar de ventajas como mensajería instantánea, gestión de proyectos, interfaz personalizable, carpetas compartidas, conexión en tiempo real con el servidor para acelerar la comunicación.

#### **6.1.1.6- Propuestas de migración de un servidor Proxy.**

Para proponer un plan de migración hacia un servidor proxy libre mencionaremos varios de los que existen, los cuales son: NTLMAPS, Privoxy y Squid. En base al estudio realizado y las necesidades de la institución se propone usar Squid.

El proxy-caché squid es una excelente solución para optimizar el uso del enlace a Internet y acelerar el tráfico web ya que almacena los contenidos más frecuentemente accedidos. Brinda además mecanismos muy flexibles para administrar el acceso por usuarios, equipos, URLs, tipo de contenido y demás. Soporta la utilización de distintos filtros de contenido, algunos de los cuales permiten la utilización de "listas negras" de acceso público que contienen listados de sitios clasificados por categoría (pornografía, juegos, etc. Posee herramientas de generación de reportes que permiten visualizar de forma flexible el acceso a la web, discriminando usuarios, sitios, horarios y brindando información muy detallada.

#### **6.1.1.7- Propuestas de migración de un servidor web.**

En el mundo existen varios servidores web libres, entre los cuales podemos mencionar Apache, Xitami, Thttp, PublicFile. Sin dudas Apache es uno de los servidores web más usados de todo el mundo. Aproximadamente el 50% de los servidores de la red utilizan Apache según las estadísticas históricas y de uso diario proporcionadas por Netcraft ([www.netcraft.com](http://www.netcraft.com)). El servidor Apache se basa en el modelo clásico de cliente-servidor. Algunas claves para su éxito son su modularidad, potencia y disponibilidad, de forma general al igual que todos los productos modulares consiste de una aplicación constituida por un núcleo básico que cumple con las características elementales de un servidor web y un conjunto de módulos para extender las funcionalidades del mismo, ejemplos de importantes módulos tenemos mod\_perl un interprete de perl empotrado en apache y jakarta un potente servidor de aplicaciones. [36]

Entre las propiedades y características de Apache por las cuales proponemos la migración hacia este servidor libre son su simplicidad, admite la última versión del HTTP/1.1, puede trabajar con CGI y/o con docenas de módulos nativos que existen para el mismo, muchas veces resulta útil el trabajo con módulos CGI pero otras veces por motivos de seguridad se puede usar módulos nativos, en estas cosas valdría la pena comparar elementos tan esenciales como la seguridad, la rapidez y ver que

queremos lograr, el uso de uno u otro dependerá también de la disponibilidad de los mismo para la ejecución de la tarea en cuestión; admite servidores virtuales, autenticación HTTP, cuenta con un servidor proxy, sus registros son muy personalizables y también brinda gran información sobre el estado del servidor.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente, además de que tiene soporte para SSL lo que constituye una gran ventaja desde el punto de vista de su seguridad como servidor web.

#### **6.1.1.8- Propuestas de una herramienta de trabajo en grupo.**

Como herramienta de trabajo en grupo proponemos la utilización del eGroupware, una solución libre para el trabajo en equipo que integra características tales como la gestión de calendarios, servicios de mensajería instantánea, gestión de documentos, gestión de proyectos, creación de sitios web personalizados, conexión servidores de correo electrónico, libreta de direcciones, registro de notas, gestión de recursos, wiki para la documentación online, además está desarrollado en PHP y permite la integración con cualquier gestor de bases de datos, con una programación modular, lo que lo hace extensible y altamente configurable.

Otras alternativas posibles para el uso de una herramienta de trabajo en grupo lo constituyen Kolab, Open Groupware, PHPGroupware, y otras que aunque no están integradas entre todas brindan un conjunto de facilidades para el trabajo en equipo, ellas son: Subversion, Gforge, Dotproject, Wikipedia, entre otras.

#### **6.1.1.9- Asterisk, alternativa libre, servidor de VoIP.**

Asterisk, alternativa libre, es un software, de una central telefónica, que se distribuye bajo licencia GPL. Se le puede conectar un número determinado de teléfonos, los cuales pueden realizar llamadas entre si. Funciona en Linux, BSD, Windows y OS X y proporciona todas las características normales de una PBX, como por ejemplo, llamadas, transferencias, correo de voz, entre otras.

Soporta muchos protocolos VoIP como pueden ser SIP, H.323, IAX y MGCP. Asterisk puede interoperar con terminales IP actuando como un registrador y como pasarela entre ambos. No necesita de ningún hardware adicional, para la transmisión de voz sobre IP, y para comunicarse con la red de telefonía pública se emplean dispositivos de costos bastante accesibles en comparación contra las PBXs propietarias, estos dispositivos la mayoría son fabricados por Digium que es el creador y el desarrollador primario de Asterisk.

#### **6.1.1.10- Métodos de seguridad en Servidores GNU/Linux.**

Toda red privada que usa servidores para la conexión de sus usuarios a la red y para el trabajo interno de la institución a la cual se rige, debe contar con respaldos que de forma automática permitan la seguridad e integridad de los datos e información. Cuando se habla de integridad de respaldos de datos e integridad de la información, se hace referencia a los datos de los usuarios en los servidores, e incluso en sus computadoras, garantizando que cada usuario pueda tener acceso solo a su

información, es decir que se mantenga la confidencialidad a través del uso de protocolos que proporcionen seguridad a los mismos.

Para lograr los objetivos anteriormente propuestos se ha llevado a cabo un estudio acerca de las herramientas y métodos que permiten lograr escalabilidad y flexibilidad de manera segura en el proceso de respaldo de datos e información, así como el establecimiento de medidas de seguridad que brinden la confidencialidad necesaria para llevar a cabo el trabajo continuo en las instituciones. Los mismos se mencionan a continuación:

*1- Uso de protocolos seguros y métodos para el manejo de certificados para evitar el envío de contraseñas en un texto claro sobre la red.* Se harán uso de protocolos seguros como ssh para la conexión remota a los servidores, y ssl para el manejo de certificados, el cual proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía.

*2- Copias de seguridad e integridad de los datos.* Se debe establecer un mecanismo de respaldo que se realice de forma automática. El mismo debe ser ejecutado en todos los servidores que guardan datos de los usuarios, como los servidores de transferencia de archivos y de bases de datos. Además se debe realizar el respaldo de la configuración de los servidores y de la información que brindan los mismos como es el caso de los servidores web. Es importante tener en cuenta que los procedimientos anteriormente mencionados deben ser ejecutados en horas en que el tráfico de la red sea lo mejor posible para evitar la pérdida de datos.

*3- Respaldo de los servidores.* El respaldo de servidores se debe de asegurar con un sistema de copias de seguridad y restauración de pérdidas apropiadas, con el objetivo de evitar pérdidas inmensas de datos que puedan perjudicar el funcionamiento de los mismos; se propone la herramienta bacula de amplia usabilidad.

*4- Políticas de seguridad en las computadoras clientes.* Se deben establecer políticas de seguridad en las computadoras de los usuarios de la institución para evitar que sean víctimas de ataques. Así como recomendar el uso de aplicaciones que envíen sus contraseñas de forma segura. Para ello se debe tener en cuenta el establecimiento de permisos restrictivos en archivos que pueden representar huecos de seguridad para el sistema, así como de una configuración acertada que permita el uso de la red y a su vez la seguridad de la computadora dentro de la misma.

*5- Políticas de seguridad en los servidores.* Se deben configurar los servidores para permitir el acceso a ellos y a su vez la seguridad de los mismos. Para ello es necesario tener en cuenta muchos aspectos como un esquema de particionado del mismo que cumpla con la funcionalidad del servidor, una configuración detallada de el firewall, el establecimiento de permisos restrictivos en archivos que pueden representar huecos de seguridad para el sistema y una configuración personalizada de los mismos así como permitir que las aplicaciones del servidor escuchen a través de puertos y protocolos seguros y establecer una configuración avanzada de los mismos estableciendo para ello además solo los permisos necesarios, de tal forma que se brinden los servicios deseados y se obtenga la mayor seguridad posible. También es necesario contar con sistemas de detección de intrusos que nos permitan monitorear nuestra red y sacar conclusiones acerca del tráfico de la misma, así como de la sobrecarga de los servidores.

*6- Bloqueo de los ficheros de configuración.* Se utilizará el comando `chattr` para marcar con una bandera los ficheros de configuración del sistema como inalterables y

así aumentar la seguridad y confidencialidad de los mismos.

*7- Comprobación de la integridad del sistema de ficheros.* Hacer uso de aplicaciones que permitan verificar la integridad de la información almacenada en los ficheros, como Tripwire. Para detectar cualquier cambio en el sistema de ficheros, el programa ejecuta varios checksums de todos los binarios importantes y ficheros de configuración, y los compara con una base de datos con valores de referencia aceptados como válidos. Además se recomienda que todos los scripts o binarios que ejecute el root tengan como propietario al usuario root y se almacenen en un directorio de su propiedad.

*8- Comprobación de la integridad de las contraseñas.* Utilizar programas de fuerza bruta para comprobar la fortaleza de las claves.

*9- Comprobación de la seguridad del sistema.* Emplear herramientas para detectar las vulnerabilidades como Nessus o Snort.

*10- Utilización del filtrado de paquetes para restringir el tráfico de entrada y salida.* Los firewalls de filtrado de paquetes leen cada paquete de datos que pasa dentro y fuera de una LAN. Puede leer y procesar paquetes de acuerdo a la información de la cabecera y filtra el paquete basado en un conjunto de reglas programables implementadas por el administrador del firewall. El kernel de GNU/Linux tiene una funcionalidad de filtrado de paquetes embebida a través del subsistema del kernel netfilter.

Además se propone utilizar herramientas con tecnologías de encriptación para administrar los equipos del dominio (sugerimos el uso de SSH), para evitar que la contraseña del root u otra información sensible sea enviada por la red en texto plano.

#### **6.1.1.11- Herramientas de seguridad en Servidores GNU/Linux.**

*Snort:* Es un IDS o Sistema de detección de intrusiones basado en red (NIDS). Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones que corresponden a ataques, barridos, intentos aprovechar alguna vulnerabilidad, análisis de protocolos, etc. conocidos. Todo esto en tiempo real.

Está disponible bajo licencia GPL, gratuito y funciona bajo plataformas Windows y UNIX/GNU/Linux. Es uno de los más usados y dispone de una gran cantidad de filtros o patrones ya predefinidos, así como actualizaciones constantes ante casos de ataques, barridos o vulnerabilidades que vayan siendo detectadas a través de los distintos boletines de seguridad. [37]

*Nessus:* Es la herramienta de evaluación de seguridad "Open Source" de mayor renombre, es un escáner de seguridad remoto para GNU/ /Linux, BSD, Solaris y Otros Unix. Este basado en plug-in (s), tiene una interfaz basada en GTK, y realiza más de 1200 pruebas de seguridad remotas. Permite generar reportes en HTML, XML, LaTeX, y texto ASCII; también sugiere soluciones para los problemas de seguridad. [38]

*SSH:* Una manera segura de acceder a computadoras remotas. Un reemplazo seguro para los comandos "r" (rlogin/rsh/rcp). OpenSSH deriva de la versión de ssh de OpenBSD, que a su vez deriva del código de ssh pero de tiempos anteriores a que la licencia de ssh se cambiara por una no libre. Provee de comunicaciones cifradas y

seguras entre dos hosts no confiables sobre una red insegura. También se pueden redirigir conexiones de X11 y puertos arbitrarios de TCP/IP sobre este canal seguro. La intención de esta herramienta es la de reemplazar a `rlogin`, `rsh` y `rcp`, y puede ser usada para proveer de `rdist`, y `rsync` sobre una canal de comunicación seguro. También con este protocolo puede hacerse uso de SSHFS, que es un sistemas de archivo que permite compartir carpetas remotas de un servidor de ssh; SCP para realizar copias seguras, donde los datos son cifrados durante la transferencia; y SFTP que utiliza el protocolo de transferencias de archivo seguro, utilizado normalmente con SSH para asegurar la transferencia. [39]

*Netfilter*: Es un poderoso filtro de paquetes el cual es implementado en el kernel GNU/Linux estándar. La herramienta iptables es utilizada para la configuración. Actualmente soporta filtrado de paquetes stateless o statefull, y todos los diferentes tipos de NAT (Network Address Translation) y modificación de paquetes. A su uso se puede adicionar además el de varios parches del mismo distribuido por desarrolladores del Netfilter para Path-O-Matic, lo cual le daría fortaleza a la seguridad del sistema.

*OpenSSL*: Es un esfuerzo de cooperación para desarrollar un set de herramientas robusto, de nivel comercial, completo en características, y "Open Source" implementando los protocolos "Secure Sockets Layer" (SSL v2/v3) y "Transport Layer Security" (TLS v1) así como también una biblioteca de cifrado de propósito general potente, además posee una amplia gama de los algoritmos criptográficos usados en muchísimos servidores web a nivel mundial, por lo cual es muy aconsejable poder usarlo en la entidad objeto de estudio, pues entre muchas otras funcionalidades nos ofrece Cifrado de datos, Autenticación de servidores, Integridad de mensajes, Autenticación opcional de usuario , basándose para ello en Criptografía de claves simétricas y asimétricas , códigos de autenticación de mensajes (MACs) y certificados digitales x.509 y muchísimos mas. [40]

*Tripwire*: Es un programa de computador Open Source consistente en una herramienta de seguridad e integridad de datos. Tripwire es útil para monitorizar y alertar de cambios específicos de ficheros en un rango de sistemas. Para mejor eficacia, se recomienda instalar el programa antes de haber conectado el computador por primera vez a Internet a fin de crear una base de datos de los ficheros existentes en el sistema, para poder contrastar los posibles cambios en éstos una vez conectado a la red. [41]

*Nagios*: Es un sistema de monitorización de equipos y de servicios de red, creado para ayudar a los administradores a tener siempre el control de qué está pasando en la red que administran y conocer los problemas que ocurren en la infraestructura antes de que los usuarios de la misma los perciban. Es un sistema complejo y completo en cuanto a sus características. La información es mostrada a través de la web. Monitoriza los hosts y servicios que se especifiquen, alertando cuando el comportamiento de la red no es el deseado y nuevamente cuando vuelve a su estado correcto. Con Nagios podremos saber en cada momento, que máquinas y dispositivos están encendidos o apagados, cuáles están fallando, cuáles funcionan correctamente, que servicios van bien y cuáles van mal, en resumen, sirve para mirar el estado casi en tiempo real de una red, sea grande o pequeña. Fue originalmente diseñado para ser ejecutado en GNU/Linux, pero también se ejecuta bien en variantes de Unix. Está licenciada bajo la GNU General Public License Versión 2 publicada por la Free Software Foundation. [42]

*SSHFS* : Este sistema nos permitirá montar los datos que tenemos en un servidor ssh

de forma remota en una maquina, igual que hacemos con NFS o Samba, pero evidentemente a través de SSH lo que sin dudas aumentara nuestra seguridad ya que los datos viajaran cifrados aunque será un poco mas lento, para ello necesitaremos en la maquina del cliente la instalación de FUSE (File System in User Space).

*SCP* : Nos brinda un medio para la transferencia de datos a través de la red de forma segura utiliza como base para la transferencia el protocolo SSH garantizando la seguridad.

*SFTP*: Es un protocolo para la transferencia segura de archivos utiliza al igual que SCP el protocolo SSH para la transferencia segura, es mas independiente de la plataforma que SCP por lo que esta mas extendido aunque en los sistemas UNIX y sus derivados es mas usado SCP.

## **6.1.2- Migración Parcial de Bases de Datos.**

En el mundo informatizado de hoy, las aplicaciones empresariales disponen al menos de una base de datos dentro de sus pilares base para varios usos, ya sea almacenamiento de datos de la misma empresa, persistencia de datos de la misma aplicación, etc.

Cualquiera, hoy en día, necesita una base de almacenamiento persistente. Pero existen un sin número de aplicaciones dedicadas a este aspecto de las bases de datos, las cuales se pueden clasificar en 3 grandes grupos: los DBMS libres, los comerciales y los gratis. Puesto que nuestro empeño es la migración a sistemas de software libre, centraremos la atención en los primeros.

Como ejemplos de DBMS libres tenemos a:

### **-PostgreSQL**

Sistema gestor de bases de datos libre del tipo Objeto-Relacional, liberado bajo la licencia BSD. El desarrollo de PostgreSQL está dirigido por una comunidad de desarrolladores y organizaciones comerciales ( como US.Army, la Biblioteca del Congreso de los Estados Unidos, el Laboratorio Nacional de Sandia, las compañías Loma Negra y Quilmas de Argentina , las compañías de Entel y la Superintendencia de AFPs en Chile, y una de las corporaciones financieras más grandes del mundo : Deutsche Bank), la cual es denominada como el PGDG (PostgreSQL Global Development Group). Está considerado como el mejor ODBMS libre hasta el momento(según las mismas empresas que le dan soporte además de las que lo usan como gestor de bases de datos como el mismo Deutsche Bank ).

Es un DBMS compatible con Oracle y DB2. Está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas. (43) (44) (45)(46)

### **-MySQL**

Es un sistema de gestión de base de datos relacional, multihilo, multiusuario con más de seis millones de instalaciones en todo el mundo, o sea, grandes empresas o las mismas PYMES lo usan mucho en sus aplicaciones de almacenamiento de datos, como por ejemplo: Motorola, la cual tiene una de las bases de datos más grandes del

mundo.

Al contrario de proyectos como el Apache, donde el software es desarrollado por una comunidad pública, y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Recientemente comprada por SUN Microsystems.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet.

Características técnicas:

- Escrito en C y en C++
- Probado con un amplio rango de compiladores diferentes
- Usa GNU Automake, Autoconf, y Libtool para portabilidad.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso completo de multi-threaded mediante threads del kernel. Pueden usarse fácilmente múltiples CPUs si están disponibles.
- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia.
- Un sistema de reserva de memoria muy rápido basado en threads.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas. (47)

## **-FireBird**

**Firebird** es un sistema de administración de bases de datos relacional (o RDBMS) SQL de código, fue reescrito de C a C++ .Firebird es una versión de la base de datos Interbase de Borland que salió con licencia OSS. Existe el proyecto de añadir capacidad de base de datos a OpenOffice.org usando Firebird pero está en una etapa inicial.

Características principales de FireBird

- Soporte completo de Procedimientos Almacenados y Triggers
- Las Transacciones son totalmente ACID compliant
- Integridad referencial
- Arquitectura Multi Generacional
- Muy bajo consumo de recursos
- Completo lenguaje para Procedimientos Almacenados y Triggers (PSQL)
- Soporte para Funciones externas (UDFs)
- Poca o ninguna necesidad de DBAs especializados
- Prácticamente no necesita configuración - isólo instalar y empezar a usarla!
- Una gran comunidad y muchas páginas donde conseguir buen soporte gratuito

- Opción a usar la versión embebida - de un solo fichero - ideal para crear CDROM con catálogos, versiones de evaluación o monousuario de aplicaciones
- Docenas de herramientas de terceros, incluyendo herramientas visuales de administración, replicación, etc.
- Escritura segura - recuperación rápida sin necesidad de logs de transacciones
- Muchas formas de acceder a tus bases de datos: nativo/API, driver dbExpress, ODBC, OLEDB, .Net provider, driver JDBC nativo de tipo 4, módulo para Python, PHP, Perl, etc.
- Soporte nativo para los principales sistemas operativos, incluyendo Windows, GNU/GNU/Linux, Solaris, MacOS.
- Backups incrementales
- Disponible para arquitecturas de 64bits
- Completa implementación de cursores en PSQL. (48)

## -SQLite

**SQLite** es un sistema de gestión de bases de datos relacional compatible con ACID, y que está contenida en una relativamente pequeña (~500 Kb) librería en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

A diferencia de los sistemas de gestión de base de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la librería SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

En su versión 3, **SQLite** permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB. (49)

*TODAS LAS ESPECIFICACIONES TÉCNICAS RELACIONADAS CON LA MIGRACIÓN DE ESTE TIPO DE HERRAMIENTAS, PUEDEN SER CONSULTADAS EN EL DOCUMENTO OFICIAL QUE SE EMITIRÁ ADJUNTO AL PRESENTE.*

## **6.2- Migración Parcial 2 Instalación de aplicaciones libres en el entorno propietario.**

Como se ha explicado anteriormente, se hace con el objetivo de reducir la resistencia al momento del cambio de la plataforma, basado en la idea de que se observan en el nuevo entorno, aplicaciones que ya son conocidas con anterioridad, resultan familiares.

Muchas son las aplicaciones libres que pueden ser utilizadas en el entorno privativo. Su número va en aumento constante. He aquí una selección de las principales:

Finalidad	Software Privativo	Software Libre
Editor de diagramas	MS Visio	Dia
Ofimática	MS Office	OpenOffice.org

Gestión de finanzas	MS Money	GNUCash, Grisbi
Gestión de proyectos	MS Project	DotProject
Maquetación	FreeHand, Adobe Illustrator	Scribus
Agenda y calendario	MS Outlook	Mozilla Sunbird
Cliente de correo	MS Outlook	Mozilla Thunderbird
Navegadores	Internet Explorer	Mozilla Firefox
IRC	Xchat	XChat
Mensajería instantánea	Pandion, MSN	Pidgin, aMSN
Bases de datos	MS SQL	PostgreSQL, MySQL
Desarrollo .NET	Visual Studio .NET	Mono
Desarrollo C/C++	C++ Builder	Code::Blocks
Desarrollo python	WingIDE	SPE
Editores	Notepad	Scite
UML	Rational Rose	ArgoUML
Gráficos 3D	3DMax	Blender
Gráficos 2D	Adobe Photoshop	Gimp
Herramienta CAD	AutoCAD	FreeCAD
Fuentes	Arial	Bitstream Vera, DejaVu
Video y multimedia	Windows Media Player	Mplayer, VLC, Totem
Gráficos infantiles	Paint	TuxPaint
Antivirus	Nod 32, Kasperky	ClamWin

### **6.3- Migración Parcial 3 de Escritorios y Entornos de Trabajo.**

Durante esta fase de la etapa de migración parcial se procederá a la sustitución del sistema base privativo por el libre.

#### **6.3.1- Softwares equivalentes.**

##### **6.3.1.1- Oficina.**

##### **Open Office, la nueva suite de herramientas ofimáticas.**

OpenOffice.org es posiblemente el programa más ilustrativo del grado de madurez que el software libre ofrece hoy día, no es una aplicación destinada a programadores, administradores de sistema u otras minorías, de conocimientos informáticos avanzados.

OpenOffice.org es un sistema de grandes dimensiones, funciones y complejidades similares a las de su rival de Microsoft. Hay otras opciones más sencillas y ágiles que pueden cubrir las necesidades ofimáticas de un buen número de usuarios, especialmente en equipos con un hardware antiguo o limitado. Un buen ejemplo es GNOME Office, un paquete de aplicaciones formado por el procesador de textos AbiWord, la hoja de cálculo Gnumeric y el gestor de base de datos GNOME-DB.

La suite está conformada por cinco sistemas que están interrelacionados.

- El procesador de textos **Writer**, la pieza angular de las letras.
- La hoja de cálculo **Calc**, la pieza angular de los números.
- El diseñador vectorial **Draw**, la pieza angular de los gráficos.
- El creador de presentaciones **Impress**, la herramienta para resumir y exponer.
- El gestor de bases de datos **Base**.

La integración entre componentes es total, ofreciendo todos en una misma interfaz básica. La reutilización de elementos generados por una aplicación por parte de otra es enorme. Se pueden crear gráficas estadísticas de datos introducidos, que luego son insertadas en documentos de texto, podemos crear efectos y animaciones para ser utilizadas en presentaciones gráficas y mucho más.

Además de esto se elige esta suite ofimática porque:

- Open Office se puede ejecutar sobre Windows lo que permite que durante el proceso de migración los usuarios de Windows lo pueden usar e irse acostumbrando a la nueva suite.
- Tiene la mejor integración con los formatos de Microsoft.
- Es la alternativa actual más factible por sus prestaciones a Microsoft Office.
- Utiliza el estándar abierto ODF para el intercambio de la información.
- Permite exportar a múltiples formatos.
- Un usuario que provenga de Microsoft Office aprende rápidamente el trabajo con Open Office, por lo que la curva de aprendizaje es rápida.

La siguiente tabla ilustra cada elemento de Microsoft Office y su equivalente en Open Office. (50)

Microsoft Word	OpenOffice.org Writer
Microsoft PowerPoint	OpenOffice.org Impress
Microsoft Excel	OpenOffice.org Calc
Microsoft Access	OpenOffice.org Base

Kpdf, alternativa libre al costoso Acrobat Reader.

Kpdf, es una herramienta libre completamente, que logra realizar las mismas funcionalidades del software de Adobe, teniendo como costo cero y ahorrando muchos más recursos de hardware. Es un sencillo lector de documentos de formato portable (PDF) que incluso puede configurarse e integrarse con *festival* (otro software de GNU/LINUX libre) y permitir leer el documento actual que se tiene abierto soportando varios idiomas. Se puede usar alternativamente también aplicaciones como *evince* y *xpdf*.

*Evince*, es una herramienta libre completamente, la cual permite leer documentos de formato PDF y postscript, actualmente viene preinstalado en las distribuciones de GNU/Linux como es el caso de Ubuntu, permite copiar textos y respaldar las imágenes presentes en los documentos. [51]

*Evince*, en versiones inferiores a la 2.20 posee algunos problemas para manejar la transparencias en documentos PDF.

La otra alternativa real corresponde a xpdf para maquinas con pocos recursos de hardware.

Mozilla Thunderbird, lector de correos libres.

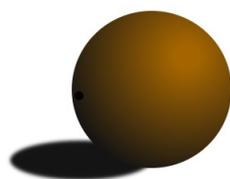
Mozilla Thunderbird es la herramienta que permitirá seguir revisando el correo. Permite etiquetar mensajes de correo electrónico colocándole a estos, distintas marcas. Incorpora: un histórico de mensaje para hacer más fácil la navegación, búsqueda con sobresaltado de la sintaxis, sistema de notificación de mensajes, integración con los servicios de correo más populares y protección del buzón contra los correos basura (Spam). Un administrador de complementos y temas, son sus características más importantes. Esta herramienta libre permitirá de forma general revisar el correo electrónico de una forma estable y segura.

No se descarta le uso de Evolution por parte de aquellos clientes que posean ordenadores con excelentes prestaciones. Este último, logra la integración con otras herramientas de mensajería instantánea. Es válido su uso en Gnome y KDE, aunque en el primero de estos consume bastante recursos. (52)

### 6.3.1.2- Gráficos.

#### **Gimp, el nuevo editor de imágenes.**

El programa de manejo de imágenes GIMP (Programa de Manipulación de Imágenes Gráficas), es sin duda alguna otro de los programas insignias del software libre, entre sus principales ventajas está que no posee costo alguno contra el costoso Photoshop que alcanza la cifra de los 4139,88 USD en su versión CS2. La otra es que, como todo producto de software libre, su código está abierto y es posible agregarle componentes de forma personalizada para lograr algún objetivo es específico, GIMP además de ser un editor de imágenes de mapa de bits, es un editor de vídeos también, no siendo esta ultima funcionalidad muy usada actualmente. Integra también en su versión actual, una extensión para la creación de gráficos vectoriales. Entre su mayor desventaja está que no presenta soporte CYMK, pero vale decir que este es usado esencialmente en la impresión en tinta, Gimp es usado por grandes empresas como Dreamworks en la producción de las película Shrek y Antz, existe en internet mucha documentación para el trabajo con esta herramienta, en general Gimp permitirá la creación y manipulación de las imágenes para lograr gráficos de alta calidad y precisión. (53)



#### **The Gimp**

QCAD, la alternativa libre a AutoCAD.

La selección de una herramienta

equivalente al AutoCad para GNU/LINUX libre es un tema complejo, existen muchas herramientas pero ninguna presenta un acabado como el CAD de Windows, QCAD es una herramienta libre basada en Qt y elaborada para el entorno de escritorio KDE, permite realizar las mismas operaciones de AutoCad pero es un poco más complejo de trabajar, no permite gráficos tridimensionales y no presenta espacios de papel. Aunque es bastante difícil recomendar una herramienta CAD para GNU/Linux, de las que existen, la de características más cercanas a Autocad es Qcad. Es de libre determinación por parte de los usuarios usarla o esperar a que aparezcan nuevas versiones del producto. [54]

### 6.3.1.3- Herramientas y utilidades.

#### La alternativa a IDEAL Administration.

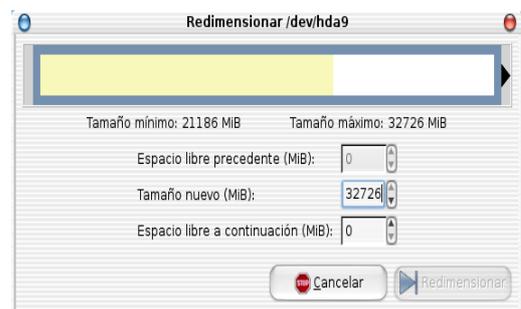
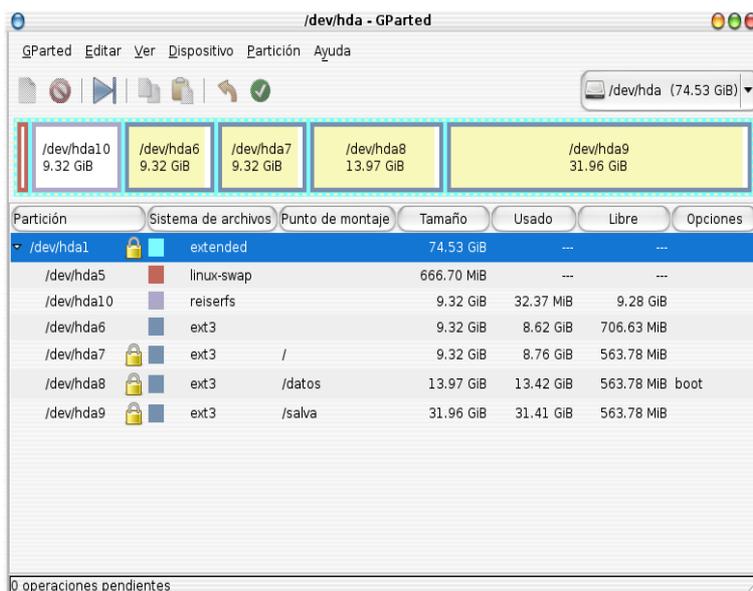
GNU/LINUX es todo un sistema operativo donde sus mayores fortalezas están inclinadas hacia el trabajo con la red y los servicios. La administración remota de sistemas y el control de los mismos se puede hacer de disímiles maneras, tanto gráficas como a través de una terminal. IDEAL Administration, permite la gestión y administración de escritorios remotos desde una computadora de la red. En GNU/LINUX existen muchas herramientas destinadas a este fin, se propone entonces usar VNC para la administración remota de las máquinas en modo gráfico y usar SSH para la administración de las máquinas a través de una terminal pudiendo este último método combinarse con gnome-terminal para lograr una conexión a través de una interfaz en modo texto ó con las combinación de gnome-terminal + Midnight Commander (mc) para el acceso a través de interfaz en ncurses. [55]

Gnome-terminal, para conexión remota a otra estación con servidor OpenSSH.

Para la conexión a estaciones de trabajo con servidores OpenSSH, puede usarse gnome-terminal, una herramienta libre que viene instalada en Ubuntu y Debian por defecto, esta herramienta es la encargada de la comunicación con el interprete de comandos del sistema, la misma permitirá conectarnos a otras maquinas sobre el protocolo SSH, bien puede hacerse una conexión en modo texto o utilizando la interfaz de ncurses a través del mc, también se podrán hacer conexiones con scp y sftp al servidor OpenSSH. [56]

Gparted, alternativa libre para Partition Magic.

El Partition Magic es un programa que permite agregar, borrar y editar particiones FAT, NTFS, EXT2 Y EXT3 en el disco duro. También permite mover archivos de un lado hacia otro, agrandar o achicar particiones, e incluso unir, pero no es libre y además es muy costoso, como alternativa libre se propone Gparted, un sistema que permite ejecutar las mismas funcionalidades que el anterior e incluso permite el trabajo sobre las particiones de Mac, ReiserFS, UFS y XFS de manera muy sencilla. (57)



Bacula, para la realización de copias de seguridad de datos.

Como herramienta para la creación de respaldos y restauración de datos

proponemos Bacula es un programa para hacer copias de seguridad muy potente que apenas tiene nada que envidiar a sus equivalentes en

software privativo.

Es una colección de demonios o servicios que se ejecutan en segundo plano (background), que cooperan entre sí para realizar copias de respaldo de los archivos necesarios, entre las principales características de esta poderosa herramienta están : el soporte de múltiples plataformas, ya que puede hacer respaldos de maquinas que estén sobre Windows, la posibilidad de usar cintas o discos para hacer los backups de datos, muy configurable y permite incluso la ejecución de respaldos o trabajos en paralelo, soporte en la instalación para bases de datos MySQL, PostgreSQL y SQLite, herramientas gráficas para la administración, seguimiento y configuración, maneja adecuado de cintas : el sistema es capaz de pasar de un cinta a otra si la anterior no posee espacio para hacer las respaldos, tiempo de vida de un respaldo en el servidor de copias de seguridad, posee encriptación sobre algoritmos como md5 para la seguridad del respaldo, permite la realización de copias de seguridad de datos incremental , diferenciales y completos, permite la restauración de datos ante desastres o pérdidas de los mismos e integrado con mondo (otra herramienta libre) permite hacer respaldos del sistema, otra alternativa importante lo constituye Amanda. [58]

K3b una completa herramienta para el quemado de Cds y DVDs.

Nero en cualquiera de sus versiones integra un conjunto de herramientas que permiten el trabajo íntegro para la creación de CDs y DVDs de datos, audio y vídeos, como alternativa libre se propone entre las más usadas K3b y GnomeBacker siendo la primera opción la más completa, permitiendo una mejor configuración y mayor cantidad de opciones. (59)

Virtual Box, la alternativa libre del Vmware.

Para el proceso de virtualización de sistemas en Software Libre la mejor alternativa presentada en estudios y pruebas en caliente a resultado ser Virtual Box, un sistema libre creado a partir de VMWare, entre sus características más sobresalientes están la modularidad, máquina virtual en descripciones XML, trabajo con carpetas compartidas, controlador USB virtual, escritorio remoto y muchas más. (60)

Gnome-btdownload, el cliente torrents de gnome.

Para la descarga de torrents es necesaria la utilización de clientes que permiten esta gestión, Bitconnect es un cliente privativo que permite esta tarea, para su sustitución se ha elegido a gnome-btdownload, esta herramienta está integrada al entorno gnome y cuando este es instalado ya trae a la misma en muchas ocasiones, además de ello en distribuciones como Ubuntu ya esta preinstalado, entre las principales características se encuentran la licencia GPL, es capaz de descargar hasta 8 torrent simultáneamente, está libre de malware y programas dañinos, permite realizar el proceso de descarga de manera muy sencilla, otras alternativas son ktorrent, freeloader, bittornado y transmission. (61)

FileZilla Client, un cliente FTP libre.

Al igual que en el sistema Windows se utilizará FileZilla Client para la conexión a servidores FTP. Sustenta FTP y SFTP sobre SSL y entre sus características se encuentran el administrador de sitios, registro de mensajes, y la cola de transferencias, permite guardar al usuario un grupo de servidores FTP y sus datos de conexión, presenta interfaz de tipo árbol jerárquico lo que hace muy sencillo el trabajo con esta herramienta permitiendo modificar tanto los datos del servidor (Si se poseen permisos en la conexión) como los datos locales. (62)

#### 6.1.3.4- Desarrollo.

Umbrello, una alternativa para el modelado de UML.

Para el diseño de diagramas UML y modelado del ciclo de vida del software existen muchas herramientas, pero realmente pocas poseen los estereotipos y diagramas completos para modelar dicha actividad como Rational Rose o Visual Paradigm, siendo este último una alternativa viable para el trabajo sobre GNU/LINUX pero no es libre y para usarla hay que desembolsar una buena cantidad de dinero al igual que las otras propietarias ya conocidas. Por otra parte, está la herramienta Umbrello, que es de las más acabadas entre todas las herramientas libres existentes hasta hoy, alguna de sus principales características son: la posibilidad de modelado de diagramas como la vista de clases, vista lógica, vista de componentes, vista de casos de uso entre otras, además permite exportar el código fuente de las aplicaciones modeladas a diferentes lenguajes como Java, C++, SQL, entre otros. (63)

Eclipse, el IDE libre de programación en Java.

Eclipse es una plataforma de desarrollo de software escrito en Java y es Software Libre, que permite el desarrollo de software de forma rápida y ágil brindando grandes facilidades, actualmente el desarrollo de Eclipse es potenciado por IBM y la comunidad de Software Libre a través de la fundación Eclipse, esta plataforma puede usarse en combinación con la maquina virtual escrita por el proyecto GNU (gcj) para lograr trabajar y programar aplicaciones libres, para la instalación solamente instale gcj de los repositorios con todas sus dependencias e instale posteriormente eclipse-gcj. (64)

Gambas 2 para migrar las aplicaciones escritas en Visual Basic 6.0.

Visual Basic es el lenguaje de programación más usado a nivel corporativo, donde la mayoría de las aplicaciones han sido escritas en este lenguaje. El proyecto gambas brinda la posibilidad de portar las herramientas desarrolladas en este lenguaje a GNU/Linux, aunque vale decir pueden existir algunas dificultades en la migración, Gambas es la herramienta de mayor alcance para realizar este proceso, en la documentación de gambas presente en esta dirección (<http://www.gambasdoc.org/help/doc/diffvb?es>) existen algunos elementos que no se pueden portar, en cambio tiene en si un conjunto de características en su diseño que han tomado lo mejor de VB y de Java lo que lo hace mas potente, un ejemplo es la eliminación de **GOTO** para capturar errores y la incluso de **try-catch-finally**, a la hora del proceso de migración es necesario tener en cuenta estas nuevas características, probablemente haya que cambiar elementos en el código. (65)

#### 6.1.3.5- Antivirus.

ClamAV, alternativa antivirus.

Como alternativa al software antivirus Nod32 se propone ClamAV, uno de los pocos antivirus de código abierto, bajo licencia GPL, recientemente comprado por la empresa Sourcefire<sup>1</sup>. Se le recomienda su uso solo en servidores especialmente de correos para analizar adjuntos y elementos maliciosos que pueden enviarse por canales de servicios telemáticos. En los clientes es poco común el uso de antivirus ya que por el diseño de la arquitectura del sistema de ficheros en GNU/Linux, es muy improbable la existencia de virus y programas malignos, el sistema de permisos que usa donde

---

<sup>1</sup> <http://www.sourcefire.com/>

una aplicación para ejecutarse deberá poseer el permiso de ejecución del usuario, es su mayor fortaleza, otra característica que le da robustez al sistema es la estructura que poseen los ficheros binarios (ELF), la cual es un tanto mas compleja de ser blanco de ataque de los programas malignos . Existen virus para el mismo pero estos no son una amenaza para el sistema, si se desea usar un software antivirus para el escritorio podría usarse ClamAV ejecutándose este mediante la línea de comandos o mediante la interfaz ClamTk la cual permite muchas facilidades para escanear nuestra computadora. ClamAV incluye un escáner que detecta más de 39.000 virus, gusanos y troyanos, y cuya base de datos se actualiza automáticamente por Internet, su uso puede ser tanto para archivos de sistemas operativos Microsoft, como para archivos de GNU/Linux. (66)

### **6.1.3.6- Seguridad.**

Con vistas a asegurar la información de los clientes durante el proceso de migración se hace necesario habilitar un servidor para las copias de los datos de los usuarios, el cual tendrá que tener suficiente cuota de almacenamiento como para poder salvar los datos esenciales de los usuarios mientras se instala el nuevo sistema, esto tendrá que utilizarse en dos momentos , inicialmente durante la instalación de las máquinas con doble booteo y posteriormente para pasar la imagen con la imagen de GNU/LINUX solamente en la segunda etapa del proceso.

Para lograr una seguridad adecuada en los clientes de la empresa se proponen las siguientes configuraciones.

**Servidor OpenSSH.**

Estará instalado en las máquinas de los clientes ya que mediante este servicio los administradores de la entidad brindarán el soporte técnico necesario a las máquinas de la instalación.

Editar el fichero `sshd_config`, dentro de `/etc/ssh`, y modificarlo para aumentar la seguridad así:

`ListenAddress 192.168.0.1`

Hacer que el ssh escuche solo la interfaz dada, sólo en un caso de que haya más de una interfaz de red (y no necesite un ssh disponible sobre éste) o que en un futuro agregue una nueva tarjeta de red (y no necesite una conexión desde ssh en ésta). (67)

`PermitRootLogin No.`

No permitir al root entrar tanto como sea posible por SSH. Si alguien quiere volverse root por vía ssh, dos logins serán necesarios y la contraseña root no puede ser obtenida a fuerza bruta por vía SSH.

`Listen 666.`

Cambiar el puerto de escucha de tal manera que el intruso no pueda estar completamente seguro de si está corriendo un demonio de sshd.

**`PermitEmptyPasswords no.`**

Evitar que los usuarios puedan iniciar sesión con contraseñas en blanco.

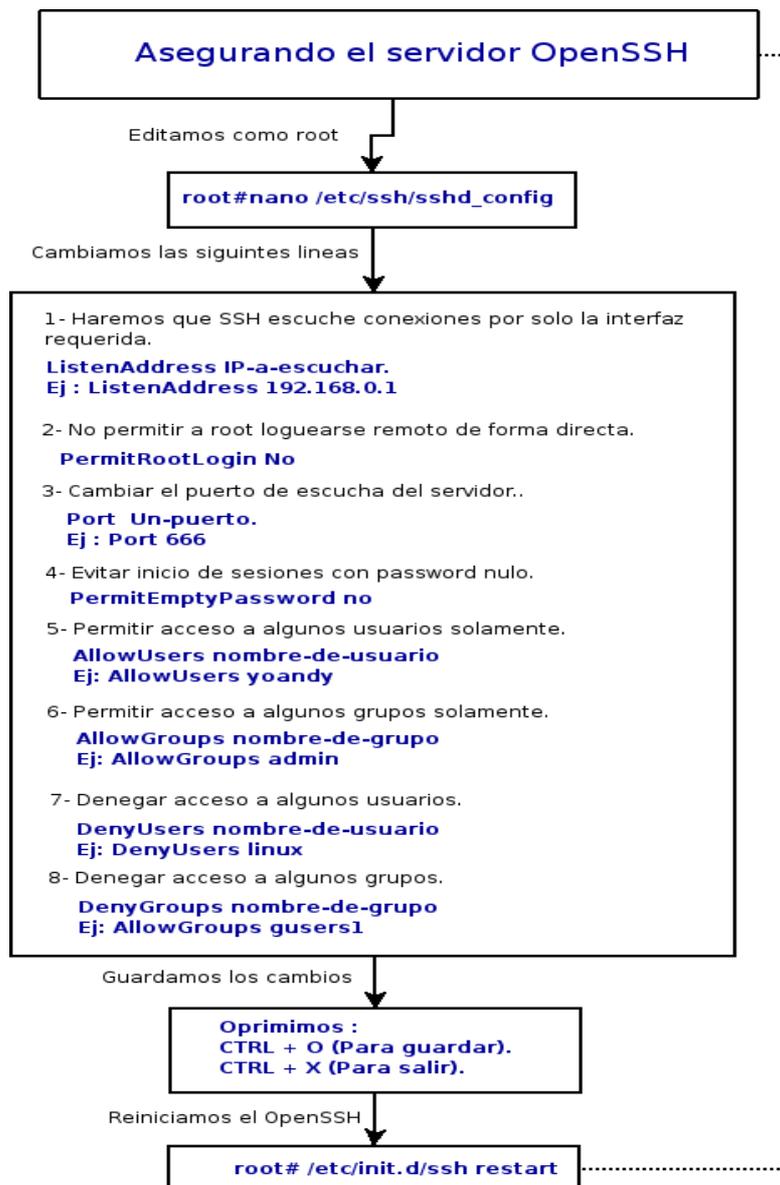
**`AllowUsers yoandy.`**

Restringir que solo algunos usuarios (los deseados, por ejemplo: miembros del equipo de soporte técnico) puedan acceder a las máquinas clientes tengan acceso a las máquinas vía SSH

### AllowGroups wheel admin.

Permitir que solamente los miembros de ciertos grupos tengan acceso vía a ssh a esta máquina. AllowGroups y AllowUsers tienen directivas equivalentes para denegar el acceso a una máquina. Las directivas para denegar que determinados grupos y/o usuarios no accedan se denominan predeciblemente "DenyUsers" y "DenyGroups".

Otra característica importante y buen consejo para la seguridad de los clientes es no permitir el uso de telnet/telnetd ya que este utiliza intercambio de información sin cifrar, por lo que se recomienda el uso de OpenSSH



Proteger el grub de accesos no autorizados. Para proteger el grub de accesos no autorizados, algo que es totalmente posible e incluso una persona con

acceso físico a una computadora sin un grub protegido si posee suficientes conocimientos puede llegar a apoderarse de la computadora por lo que es sumamente importante protegerla, para ello en el fichero /boot/grub/menu.lst agregamos la línea siguiente si no existe, si existe y está comentada

simplemente se le quita el comentario y se le pone el valor correspondiente a la contraseña encriptada:

```
password --md5 $1$bw0ez$t1jnxxKlfMzmnDVaQWgjP0
```

El valor encriptado se logra con la aplicación grub-md5-crypt, la cual genera este valor tecleando en la consola este comando y suministrado una contraseña, el valor devuelto lo colocamos en la línea nombrada anteriormente y listo, esto nos garantizara que nadie pueda editar el grub de nuestra estación de trabajo.

## **BIOS**

Es importante establecer una contraseña en la BIOS para prohibir la carga del sistema a través de un disquete cambiando la configuración de su BIOS. Esto les cerrará el paso a las personas no deseadas para que puedan cargar el sistema GNU/Linux con un disco especial de inicio y lo protegerá de las personas que traten de cambiar las características de la BIOS como lo son iniciar el sistema de un disquete o cargar el servidor sin preguntar la contraseña.

Proteger la cuenta root.

Para evitar que la cuenta de root se quede abierta por tiempo indefinido por error y que algún usuario sin el debido acceso tome o afecte información importante configuremos el sistema para que el usuario root al cabo de un tiempo salga de su sesión del sistema, para ello editemos el fichero /etc./profile y agreguemos la línea siguiente debajo de la línea que mostramos a continuación.

```
"HISTFILESIZE="  
TMOUT=3600
```

Deshabilitemos los servicios innecesarios.

Para ello instalaremos la aplicación rcconf y eliminemos los servicios que están innecesariamente ejecutándose.

### **6.1.3.7- Aplicaciones Críticas.**

Se llaman aplicaciones críticas, a aquellas que no tienen un equivalente desarrollado todavía en el mundo del software libre. Son por lo general, aplicaciones de propósito específico. Para realizar la migración de dichas aplicaciones es necesario desarrollarlas completa o parcialmente usando tecnologías libres. Son candidatos para esto Java, C++, C#, y PHP 4.x o 5.x según los estudios que se realicen más adelante con los IDE Monodevelop, Quanta, Kdevelop, CodeBlock, Eclipse, Gambas y NetBean.

Se tiene previsto una migración parcial por entidades en dependencia de su relación interna con otras. Se visitaría cada entidad buscando posibles soluciones para evitar el desarrollo desde un comienzo de cualquier aplicación y si estas no existen, entonces se procedería a desarrollarlas.

Se incluyen en este grupo, los sistemas de gestión y de cualquier otra índole, adquiridos mediante empresas desarrolladoras y comercializadoras de software o elaborados por el personal técnico de la institución.

## 6.3.2- Distribuciones a utilizar en Escritorios y Servidores.

### 6.3.2.1- Nova: Distribución cubana de Software Libre.

La idea de crear una distribución de GNU/Linux surge de un grupo de estudiantes de la Universidad de las Ciencias Informáticas, como respuesta a la necesidad de una plataforma que garantizara la compatibilidad del software que se estaba desarrollando, con los sistemas libres que tanto auge tienen en el mundo actual. Posteriormente el proyecto se volvió más ambicioso con la inminente migración a software libre de algunos organismos estatales y ministerios, convirtiéndose en una plataforma para generar distribuciones a la medida.

En estos momentos Nova tiene como objetivo la creación de un sistema operativo, no la mera personalización de una distribución. Se aspira a proveer un producto orientado a usuarios inexpertos que hayan tenido que migrar de Microsoft Windows a entornos GNU/Linux o cuya experiencia con computadoras sea nula. Se pretende automatizar la mayor cantidad de procesos posible de forma que la interacción de la persona con el sistema sea fácil e intuitiva facilitando el proceso de transferencia de conocimientos y aprendizaje, algo tan difícil cuando se trata de asimilar nuevas tecnologías. (68)



Debido a los apenas 3 años del proyecto Nova, su utilización es relativamente reciente en algunas instituciones interesadas en migrar de forma paulatina a software libre.

La probabilidad de infección del sistema NOVA por parte de programas maliciosos es casi nula, incluso sin tener instalado un antivirus, por tanto no se daña el sistema por esta causa tan frecuentemente vista haciendo uso del software privativo. En cuanto al consumo y falta de espacio para la instalación o compilación de programas, no sería una preocupante a tener en cuenta debido a que la mayor parte de los programas o software necesarios ya están pre-compilados, y se llevan una menor capacidad de almacenamiento a la hora de ser instalados. A todo lo anteriormente mencionado se le debe sumar que las aplicaciones no requieren supervisión tan estrecha ni pagos de pólizas de mantenimiento necesarias para obtener las actualizaciones de los productos (Service Packs); los cuales pueden contener actualizaciones para la estabilidad del sistema, la compatibilidad del programa, la seguridad, etc.

El sistema operativo NOVA por ser un sistema libre presenta las características de los mismos, siendo por ello muy robusto, estable y rápido: Ideal para servidores y aplicaciones distribuidas. A esto se añade que puede funcionar en máquinas humildes: alcanza correr servicios en un x86 a 200 MHz con calidad. Presentando además la posibilidad de modificación y la variedad de programas que se pueden seleccionar en Internet o en el repositorio de la distribución, de acuerdo a las necesidades del cliente.

La distro no está restringida a personas con grandes conocimientos de informática, los desarrolladores de la misma han hecho un gran esfuerzo por dotar el sistema de asistentes de configuración y ayuda, además de un sistema gráfico muy potente y cómodo para los usuarios. (69)

Nova presenta un Live-USB, Similar a un Live-CD, pero con la característica que funciona íntegramente desde una memoria flash, con acceso lectura-escritura, lo que permite al usuario almacenar todos sus documentos y configuraciones personales. De esta forma cualquier usuario de la misma puede llevar su propio sistema operativo en un bolsillo. Lo que sería sumamente cómodo y eficiente a la hora de trabajar tanto en el centro laboral, el hogar, lejos del mismo, etc.

Cuenta además con un Centro de Control de Herramientas, que funciona como una capa de presentación a todas las aplicaciones de configuración y administración, muy similar al panel de control de Windows, estando diseñada para que un usuario común y corriente encuentre lo que busca sin mucho esfuerzo. Está disponible en su versión 1.0.

Al ser una plataforma estable y segura, es favorecido también el desempeño de aplicaciones de todo tipo tales como: bases de datos, aplicaciones XML, multimedia, la rápida navegación por la Web, y la velocidad de las aplicaciones, muy superiores a los sistemas Windows.

### **6.3.2.2- Debian GNU/Linux .**

Debian es un sistema operativo libre, para su computadora, que utiliza el núcleo Linux (el corazón del sistema operativo), pero la mayor parte de las herramientas básicas vienen del Proyecto GNU; de ahí el nombre GNU/Linux. Viene con más de 8710 paquetes, programas precompilados distribuidos en un formato que hace más fácil la instalación en su computadora.

- Contiene mas software que cualquier otra distribucion comercial.
- Mejor sistema de paquetería. Fácil actualización, descentralización (elegimos que instalar y que no), Sistema operativo completo, a la medida.
- Distro mas estable.

**Esta mantenido por sus usuarios:** Si algo necesita ser arreglado o mejorado, simplemente lo hacemos.

**Soporte incomparable:** El correo enviado a las listas de correo ([http://www.debian.org/support#mail\\_lists](http://www.debian.org/support#mail_lists)) frecuentemente obtiene respuesta en 15 minutos (o menos), gratuitamente, y por las personas que lo desarrollaron. Compare esto al típico soporte telefónico: horas gastadas en el teléfono, pagando dinero, sólo para tener a alguien que no conoce el sistema lo suficientemente bien como para entender su pregunta.

**No estará solo en su elección :** Un amplio abanico de organizaciones e individuos usa Debian. En la siguiente página Quién está usando Debian? <http://www.debian.org/>

users/ si desea una descripción de algunos sitios de perfil alto que usan Debian, y han decidido enviar una corta descripción de cómo hacen uso de Debian, y por qué.

**El mejor sistema de empaquetamiento de software del mundo:** Cansado de viejos archivos de software tres versiones anteriores al actual desordenando su sistema? O de instalar software sólo para encontrar que hace que su sistema se colapse debido a conflictos de software? Dpkg, el sólido sistema de empaquetamiento de Debian, se encarga de estos asuntos por usted.

**Instalación sencilla:** Si ha oído que Linux es difícil de instalar, entonces no ha probado Debian últimamente. Se está mejorando constantemente el proceso de instalación. Puede realizar la instalación directamente desde DOS o un CD, discos flexibles (floppies) o incluso a través de la red.

**Increíble cantidad de software:** Debian viene con más de 8710 piezas diferentes de software (<http://www.debian.org/distrib/packages>). Cada bit de las mismas es libre. Si tiene software comercial que corre bajo Linux, usted todavía puede usarlo - de hecho, puede incluso existir un instalador en Debian que automáticamente instale y configure todo por usted.

**Paquetes bien integrados:** Debian sobrepasa a todas las otras distribuciones en lo bien integrados que están sus paquetes. Como todo el software es empaquetado por un grupo coherente, no sólo puede encontrar todos los paquetes en un mismo sitio, sino que puede estar seguro de que hemos eliminado todos los problemas al respecto de complejas dependencias. Aunque creemos que el formato deb tiene algunas ventajas sobre el rpm, es la integración entre paquetes lo que hace a Debian más robusta.

**Código fuente:** Si usted es un desarrollador de software, usted apreciará el hecho de que haya cientos de herramientas y lenguajes de desarrollo, además de millones de líneas de código fuente en el sistema base. Todo el software en la distribución principal esta conforme con el criterio de las Directrices de Software Libre de Debian (DFSG) ([http://www.debian.org/social\\_contract#guidelines](http://www.debian.org/social_contract#guidelines)). Esto significa que usted puede usar libremente este código para estudiarlo, o para incorporarlo a un nuevo proyecto de software libre. También hay cantidad de herramientas y código apropiado para el uso en proyectos propios.

**Actualizaciones fáciles:** Gracias a nuestro sistema de empaquetamiento, actualizarse a una nueva versión de Debian es muy fácil. Sólo ejecute `apt-get update` ; `apt-get dist-upgrade` y usted puede actualizarse desde un CD en cuestión de minutos o direccione `apt` a alguno de los espejos (<http://www.debian.org/mirror/list>) de Debian y actualícelo desde la red.

**Sistema de Seguimiento de Errores:** El Sistema de Seguimiento de Errores (<http://bugs.debian.org/>) de Debian está disponible públicamente. No se intenta esconder la realidad de que el software no siempre trabaja de la manera que los usuarios desean. Aconsejamos a los usuarios que envíen informes de errores y serán notificados cuando y por qué el error ha sido solucionado. Este sistema permite que Debian responda a los problemas rápida y honestamente. (\*)

### 6.3.2.3- Ubuntu.

Ubuntu es una distribución Linux que ofrece un sistema operativo predominantemente enfocado a ordenadores de escritorio aunque también proporciona soporte para servidores.

Basada en Debian GNU/Linux, Ubuntu concentra su objetivo en la facilidad de uso, la libertad de uso, los lanzamientos regulares (cada 6 meses) y la facilidad en la instalación. Ubuntu es patrocinado por Canonical Ltd., una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.

El nombre de la distribución proviene del concepto zulú y xhosa de ubuntu, que significa "humanidad hacia otros" o "yo soy porque nosotros somos". Ubuntu es un movimiento sudafricano encabezado por el obispo Desmond Tutu, quien ganó el Premio Nobel de la Paz en 1984 por sus luchas en contra del Apartheid en Sudáfrica. Mark Shuttleworth, mecenas del proyecto, se encontraba muy familiarizado con la corriente. Tras ver similitudes entre los ideales de los proyectos GNU, Debian y en general con el movimiento del software libre, decidió aprovechar la ocasión para difundir los ideales de Ubuntu. El eslogan de Ubuntu – "Linux para seres humanos" (en inglés "Linux for Human Beings") – resume una de sus metas principales: hacer de Linux un sistema operativo más accesible y fácil de usar.

La versión más reciente de Ubuntu es la 8.04 LTS (Hardy Heron), lanzada el 24 de abril de 2008.

Ubuntu es una distribución de GNU/Linux, basada en Debian que tiene las siguientes ventajas:

1. Apt, de las distribuciones que he usado las que se basan en Debian llevan ventaja gracias a esta potente herramienta, que hace mas fácil la instalación y configuración de paquetes.
2. Una comunidad muy grande, dinámica y dispuesta a colaborar, que si bien no es 100% de Ubuntu, si no mas bien de Debian y otras distribuciones basadas en esta, te hace mas fácil la vida, para iniciarte en el mundo GNU/Linux.
3. Se tiene una nueva versión aproximadamente cada 6 meses.
4. Pasar de una versión a la siguiente es muy sencillo (basta cambiar el sourceslist y hacer un apt-get dist-upgrade), esto en las otras distribuciones no es posible (o si lo es no es tan sencillo como en Ubuntu).
5. Detección de hardware muy buena, por lo general no se tiene ningún problema con Ubuntu.
6. Cada versión tiene soporte tecnico de por lo menos 18 meses desde su salida.
7. Soporte para 3 arquitecturas, x86, Amd64 y PowerPC.
8. Alrededor de 16000 programas en los repositorios de Ubuntu, y van aumentando.
9. Tiene las ultimas versiones de la mayoría de los programas, y tardan poco en incluir una nueva.
10. Si se descubre algún Bug o fallo de seguridad, las actualizaciones de seguridad salen de forma rápida.

Más información sobre esta distribución de GNU/Linux disponible en:

<http://www.ubuntu-es.org/>  
<http://www.ubuntu.com/>

### 6.3.3- Clonación de la Imagen a varias computadoras: Sistclon.

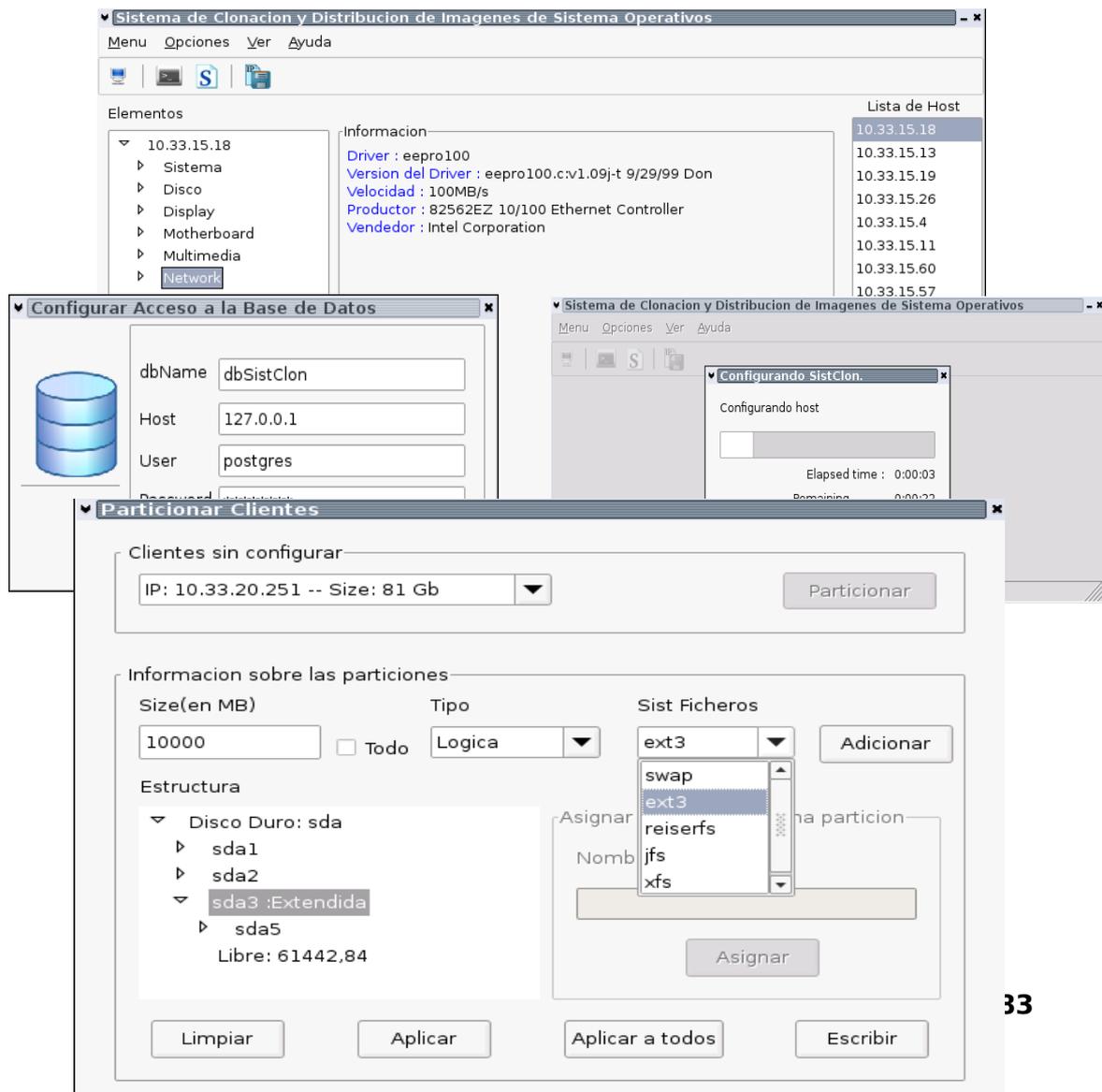
En la actualidad uno de los mayores problemas al que se enfrenta cualquier institución o empresa que presente al menos un número considerable de computadoras, es el de dar soporte y distribuir la misma imagen de sistemas operativos y software a una serie de computadoras por igual.

El proceso después de ser repetido una y otra vez a cada computadora de forma individual se convierte en tedioso. Por esta razón los desarrolladores se entregaron a la tarea de automatizar este proceso lo mejor posible para brindar rapidez y eficiencia a los administradores y técnicos que se encargan de mantener y dar soporte a dichos grupos de computadoras. (70)



No exenta de esa Universidad de las de clonación de computadoras de una subred, facilitando el acceso y control de sus dispositivos desde un servidor. Además de manipular el particionamiento de los discos duros y el proceso de instalación de la imagen.

el Software Libre en la llo de su propio sistema á controlar todas las



SistClon es un software orientado al mantenimiento y la administracion remota de un conjunto de computadoras. El mismo en su versión 0.3 (versión estable de este producto que es utilizada actualmente en el proceso de clonación de imágenes de la docencia de la UCI) es capaz de brindar soporte a diferentes tipos de motherboard.

La clonación se realiza usando unicast como método de transmisión de datagramas IP. El método de clonación empleado es la clonación semiconsciente. Puede realizar auditorias de hardware y software para sistemas GNU-Linux.

En versiones posteriores se pretende llegar al método de clonación basado en la estructura lo cual posibilitaria hacer de este producto el único software para la clonación remota completamente libre. También se pretende añadir la posibilidad de clonar usando los protocolos p2p y multicast como metodo de transmisión de paquetes añadido lo que brindaria al sistema una gran velocidad y control.

#### **6.4- Corrección de Errores.**

Es la etapa de migración parcial el momento de probar en tiempo real lo planificado por especialistas y dirigentes institucionales, por lo que sirve este período de pruebas pilotos para hacer las correcciones y ajustes necesarios tanto al Plan de migración como al Plan de acción institucional a fin de evitar problemas innecesarios durante el despliegue a ejecutarse de manera masiva en todo la institución en la siguiente etapa.

#### **6.5- Metapaquete de cursos de formación.**

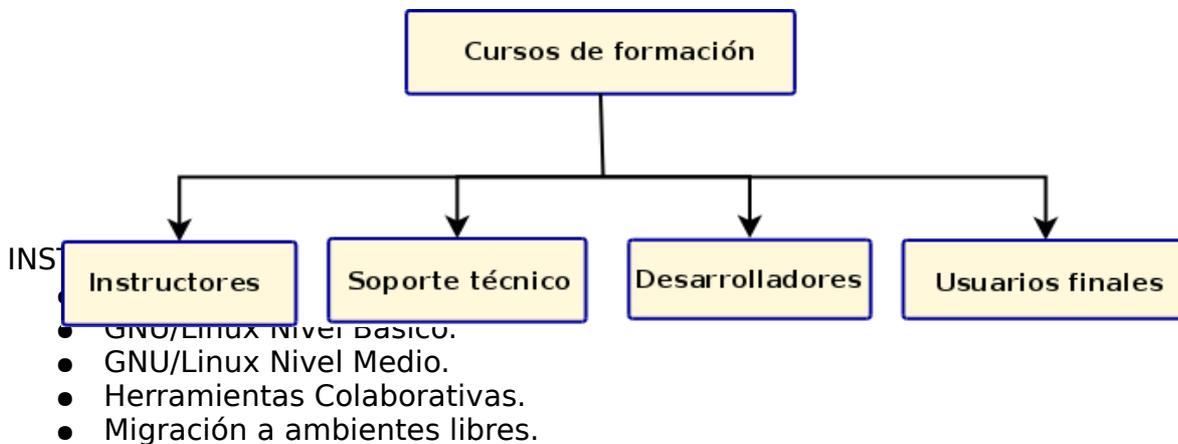
Se trata de un paquete de cursos basados en el modelo de software libre que se imparte como parte del 2do perfil de la facultad de Software Libre de la UCI. Integran el mismo los siguientes:

<b>GNU/Linux nivel 0</b>
<b>GNU/Linux básico</b>
<b>GNU/Linux medio</b>
<b>Herramientas colaborativas</b>
<b>Desarrollo con metodologías XP y SCRUM</b>
<b>Migración a ambientes libres</b>
<b>PostgreSQL avanzado</b>
<b>Programación en PERL</b>
<b>Programación en Bash</b>
<b>Programación PHP</b>
<b>Programación Python</b>

## Gestión de Proyectos con DotProject

### Servicios telemáticos.

La distribución de los cursos por perfiles queda de la siguiente manera:



#### SOPORTE TÉCNICO:

- Herramientas Colaborativas.
- Migración a ambientes libres.
- Servicios Telemáticos.

#### DESARROLLADORES:

- GNU/Linux Nivel Básico.
- GNU/Linux Nivel Medio.
- Herramientas Colaborativas.
- Desarrollos con Metodologías Xp y Scrum.
- PostreSQL avanzado
- Programación en PERL
- Programación en Bash
- Programación PHP
- Programación Python
- Gestión de Proyectos con DotProject

#### USUARIOS FINALES:

- GNU/Linux Nivel 0.
- GNU/Linux Nivel Básico.

# **Parte 7**

# **ETAPA DE MIGRACIÓN**

# **TOTAL.**

## **7- ETAPA DE MIGRACIÓN TOTAL.**

Completamente igual a la etapa anterior, salvo que en esta se despliegan las soluciones de migración a todos los locales y dependencias de la entidad siguiendo el organigrama de la misma. Es una etapa compleja por la concentración de flujos de trabajo que concurren en la misma. Su avance implica, la inclusión de un mayor número de personas, el cumplimiento exacto del cronograma de trabajo y la elaboración de la documentación de todo el proceso.

Deben seguirse con detenimiento las “disposiciones” contempladas en el Plan de Migración. Estarse atento a cualquier irregularidad que pueda presentarse y que requiera recurrir a métodos o soluciones con las que no se contaba y que no fueron detectadas a pesar de haber realizado todo un período de pruebas previamente. Resulta vital prestar atención a las opiniones de los usuarios y recopilar cualquier tipo de quejas o inconvenientes, a fin de solucionarlos cuanto antes, evitando retrasos innecesarios y la recesión del proceso.

### ***7.1- Migración Total 1 de Servidores y Bases de Datos.***

Idem a 6.1.

Tener en cuenta que de tratarse de pequeñas o medianas bases de datos, el mejor candidato es MySQL. Por el contrario, si se manejan grandes volúmenes de datos, entonces puede obtenerse por PostgreSQL.

### ***7.2- Migración Total 2 Instalación de aplicaciones libres en el entorno propietario.***

Idem a 6.2.

### ***7.3- Migración Total 3 de Escritorios y Entornos de Trabajo.***

Idem a 6.3.

# **Parte 8**

## **ETAPA DE**

### **CONSOLIDACIÓN.**

## 8- ETAPA DE CONSOLIDACIÓN.

### 8.1- Formación y Certificación de usuarios.

A lo largo del proceso de migración ha quedado evidenciada la necesidad de acometer en todo momento la formación y capacitación de usuarios, comenzando incluso desde etapas tempranas y manteniéndola aún después de concluida la misma, a fin de generar recursos para el aprendizaje y documentar todo lo realizado. Cuestiones que aunque no lo parezcan, tienen un alto valor agregado y constituyen ahorros considerables para la institución y fuente de generación de ganancias, en caso de ser usadas con esos fines.

Cada institución, en dependencia de sus características, elaborará y pondrá en práctica su propia estrategia de capacitación, dando especial prioridad a las cuestiones relacionadas con su hacer diario, así como, a las tecnologías que en un futuro inmediato se introducirán en el ambiente laboral.

Pensando más en grande, los usuarios avanzados podrían acceder a programas internacionales de certificación de usuarios, supervisados por el Linux Professional Institute. Esto permitirá a la institución ganar prestigio y posicionarse en un buen lugar en el mercado.

El programa de formación LPI de Linux Professional Institute está especialmente diseñado para proporcionar los conocimientos y habilidades necesarios para administrar cualquiera de las distribuciones de los sistemas operativos Linux y sus herramientas asociadas.

Actualmente, el programa LPIC está diseñado en dos niveles (y un tercero todavía en fase de desarrollo):

- Junior Level Administration (LPIC-1)  
(Nivel inicial de certificación)
- Intermediate Level Administration (LPIC-2)  
(Nivel medio de certificación)

Los candidatos a la certificación LPI de Linux Professional Institute deben superar un total de 2 exámenes oficiales para cada nivel de certificación:

Junior Level Administration (LPIC-1)	101
	102*
Intermediate Level Administration (LPIC-2) **	201
	202*

Los cursos pueden hacerse de manera presencial o virtual. Las pruebas si se hacen bajo la supervisión de los certificadores de la institución que lo imparte. Más información se encuentra disponible en : <http://www.nhbarcelona.com/certificaciones/linux-lpi-ruta.htm>.

## **8.2- Soporte y Mantenimiento: Servi-Desk.**

El Service Desk es una aplicación generalmente web que gestiona casi todos servicios que presta el departamento de informática, desde soporte técnico hasta compras de material informático. Esta basado en los módulos de SLAs \* y en las tecnologías de Help Desk.

El Help Desk se basa en un conjunto de recursos técnicos y humanos que permiten dar soporte a diferentes niveles de usuarios informáticos de una empresa.

- Servicio de soporte a usuarios de “sistemas microinformáticos”.
- Soporte telefónico centralizado Hotline.
- Atendido de forma inmediata e individualizada por Técnicos Especializados.
- Apoyado sobre un Sistema informático de última generación.
- Permite asignar tareas a técnicos propios o externos a su empresa.

OneOrZero HelpDesk and Task Management System es un programa de código abierto muy potente y ligero. Está desarrollado en PHP y utiliza la base de datos MySQL y Sqlite. Es rápido, personalizable y se puede ejecutar en cualquier plataforma. Se trata de un producto gratuito, tanto para uso personal como comercial.

La empresa que maneja los desarrollos se encuentra en Australia con sucursales en Gran Bretaña.

### **Requerimientos mínimos:**

- Sistema operativo Windows o Linux.
- Apache
- PHP 4.0 (o versión superior)
- MySQL 4.0 (o versión superior)
- SSL (para transacciones seguras)
- Software de foros (opcional) Se recomienda XMB Forum (<http://www.xmbforum.com/>)
- LDAP(Opcional)
- POP3(Opcional)
- Sendmail(Opcional)

El software posee Licencia GPL. Está programado en PHP y otros que son compatibles

(javascript, ajax, xml, html), por lo que brinda mas funcionalidades para futuros desarrollos. Posee traducción a Portugués, Brasileño, Danés, Holandés, Inglés, Francés, Alemán, Hebreo, Italiano, Noruego, Ruso, Español y su interfaz puede ser vista desde cualquier navegador.

En la pagina <http://www.oneorzero.com/smforum/> se encuentra una amplia disertación de temas relacionados con OneOrZero en los aspectos de instalación, configuración, traducción, soporte, desarrollo y muchos más; organizados de manera excelente por versiones, temas generales y espacios para exactamente 9 idiomaslo que da una idea de la utilización que tiene a nivel mundial este Service Desk.

OneOrZero, es un sistema muy bueno, y una solución sencilla y simple. Este sistema, incorpora varios estilos de diferentes colores, y múltiples opciones, sin duda, es una buena herramienta para la empresa pequeña y/o mediana.

Sin embargo, esta herramienta tiene muy buenas funciones, tales como un cuadro de comparaciones de tiempo de respuestas, estados, avisos, documentación y está en español. (71)

La UCI posee una personalización de este, denominada NOVA DESK.

# **ANEXOS.**

## ANEXO 1 HERRAMIENTAS PRIVATIVAS Y SUS EQUIVALENTES LIBRES.

Propósito	Estas son las herramientas propietarias.	Estas son las Herramientas Libres con las que pueden sustituirse.
Paquete Ofimático	Microsoft Office	OpenOffice.org
Visor de documentos	Acrobat Reader	Kpdf
Navegador Web	Internet Explorer	Mozilla FireFox
Paquete Macromedia	Macromedia: .Dreanweaver .Flash	. Nvu + Quanta . <b>No alternativa</b>
Diseño	Autocad	QCAD
Edición de Imágenes	Photoshop	GIMP
	Partition Magic	GParted
Desarrollo	NetBeans	NetBeans
Modelado UML	Enterprise Architect	ArgoUML
Desarrollo	Visual Basic	. Gambas . Mono 1.2.3
Quemado de CD y DVD	Nero	K3B
Anti-virus	Nod 32 Antivirus	ClamAV
Diseño 3D	3D Max Studio	Blender
Diseño gráfico (vectorial)	CorelDRAW	Inkscape Skencil
Edición de Video	Adobe Premiere	Cinelerra Jahshaka avs video converter
Gestión de Software	MSProject	Planner GantProject Dotproject Track
Programación HTML	Editores HTML Dreamweaver  FrontPage	Quanta Plus Bluefish WebMaker Screem Toppage WebDesigner ScripEditor

		Webdevelopment (plugging firefox) Amaya August Nvu Dreamweaver Emulado ZendStudio
Programación	JBuilder	Jbuilder for Linux NetBeans Sun One Studio Vide Eclipse
Programación	VisualBasic	Gambas
Programación	C++ Builder x.x	KDE Studio Gold Dev-C++ Kylix3 [Prop] vtkBuilder foxBuilder wxDesigner Code Crusader[Prop] CodeWarrior[Prop] GBuilder Source Navigator TimeStorm[Prop] Understand for C++[Prop] SlickEdit.[Prop] Vide Ultimate Anjuta + Glade Code::Blocks Eclipse + Plugin C++ Kdevelop + QtDesigner
Programación	VisualStudio.net	CodeForge Kdevelop + Qt3 Designer Eclipse Emacs Xemacs Monodevelop
Programación	Pascal	FreePascal GNU Pascal

		Rshell (Borland Pascal 7.0) Lazarus
--	--	--

**Otras aplicaciones y sus respectivos equivalentes se encuentran disponibles en [www.cdlibre.org](http://www.cdlibre.org).**

## **ANEXO 2**

# **REFERENCIAS BIBLIOGRÁFICAS**

## REFERENCIAS BIBLIOGRÁFICAS.

1. Software Libre (2007, febrero). Extraído el 13 de febrero, 2007 de: [http://es.wikipedia.org/wiki/Richard\\_Stallman](http://es.wikipedia.org/wiki/Richard_Stallman).
2. Stallman, Richard M. Software Libre para una sociedad libre. Editorial Traficantes de Sueños. España. 2004. P.99.
3. Báez Bezama, Eric (2002, noviembre). Software Libre - Porqué Linux: Una Guía Para Migrar. *Software Libre Chile*. Extraído el 8 de noviembre, 2006 de: <http://www.softwarelibre.cl/drupal/?q=node/460>.
4. Abella, A., Sánchez, J & Segovia, M. (2004). Libro Blanco de Software Libre en España.
5. Acedo, Ramón (2006, abril). Migración a Software Libre. Extraído el 8 de noviembre, 2006 de: <http://cargol.net/~ramon/migracion>.
6. Hnizdur, S. (2003, octubre). Directrices IDA de migración a Software de Fuentes Abiertas. Unión Europea. 2003.
7. Chávez Frías, Hugo. (2004, diciembre). Decreto 3390. Gaceta oficial # 38.095. Extraído el 7 de febrero, 2007.
8. Díaz, José Ramón (2004, diciembre). Nuevos Modelos de Negocio basados en Software Libre. Extraído el 22 de enero, 2007 de: <http://najaraba.com/ModelosNegocio/NuevosModelosNegocio.pdf>
9. Hernández Espinosa, Rafael (). Reflexiones sobre el uso del Software Libre en Cuba. Extraído el 22 de enero, 2007 de: <http://www.monografias.com/trabajos40/software-cuba/software-cuba.shtml>.
10. (2002, junio). Guía para la migración de Software Libre en las entidades publicas de Perú. Instituto Nacional de Estadísticas e Informática. 2002.
11. <http://www.winrar.es>
12. <http://www.tucumanlinux.com.ar/porque/diferencias.html>
13. [http://www.ciao.es/MySQL\\_\\_Opinion\\_1075588](http://www.ciao.es/MySQL__Opinion_1075588)
14. <http://www.genbeta.com/2005/09/13-zimbra-correo-y-calendario-libres-al-mas-puro-estilo-gmail>
15. <http://blackshell.usebox.net/archivo/297.php>
16. [http://es.wikipedia.org/wiki/BSD\\_\(licencia\)](http://es.wikipedia.org/wiki/BSD_(licencia))
17. [http://es.wikipedia.org/wiki/Apache\\_License](http://es.wikipedia.org/wiki/Apache_License)
18. [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro](http://linux.ciberaula.com/articulo/linux_apache_intro)
19. <http://osdir.com/ml/education.colombia.slec/2001-02/msg00082.html>
20. [http://es.wikipedia.org/wiki/Licencia\\_P%C3%BAblica\\_de\\_Mozilla](http://es.wikipedia.org/wiki/Licencia_P%C3%BAblica_de_Mozilla)
21. <http://www.microsiervos.com/archivo/ordenadores/gpl-3-software-libre.html>
22. <http://www.genbeta.com/2007/06/30-gpl-v3-version-final-se-hace-publica>
23. (2002, junio). Guía para la migración de Software Libre en las entidades publicas de Perú. Instituto Nacional de Estadísticas e Informática. 2002.
24. Comunidad Europea. “Directrices IDA de migración a Software de Fuentes Abiertas”. [octubre/2003].
25. [http://es.wikipedia.org/wiki/Microsoft\\_Office\\_Open\\_XML](http://es.wikipedia.org/wiki/Microsoft_Office_Open_XML)
26. <http://www.openxml.info>
27. Comunidad Europea. “Directrices IDA de migración a Software de Fuentes Abiertas”. [octubre/2003].
28. Comunidad Europea. “Directrices IDA de migración a Software de Fuentes Abiertas”. [octubre/2003].
29. Comunidad Europea. “Directrices IDA de migración a Software de Fuentes

- Abiertas". [octubre/2003].
30. Tesis de yoandy.
  31. Tesis mia.
  32. Instituto Nacional de Estadísticas e Informática. "Guía para la migración de software libre en las Entidades Públicas de Perú". [junio/2002].
  33. [www.cdlibre.org](http://www.cdlibre.org)
  34. [www.smaldone.com.ar/profesional/linux.shtml](http://www.smaldone.com.ar/profesional/linux.shtml)
  35. <http://dns.measurement-factory.com>
  36. [www.netcraft.com](http://www.netcraft.com)
  37. <http://www.snort.org/>
  38. <http://www.nessus.org/>
  39. <http://www.openssh.com/>
  40. <http://www.openssl.org/>
  41. <http://www.tripwire.com/>
  42. <http://www.nagios.org/>
  43. <http://www.postgresql.org>
  44. [Http://www-postgresql.cl](http://www-postgresql.cl)
  45. <http://www.postgresql.mx.com>
  46. <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/x56.html>
  47. <http://www.mysql.com>
  48. <http://www.firebird.com>
  49. **no sqlite**
  50. [http://guias-ubuntu.ourproject.org/index.php?title=Gu%C3%ADa\\_Base/Mil\\_y\\_un\\_programas\\_libres\\_y\\_gratuitos/Ofim%C3%A1tica\\_con\\_OpenOffice.org](http://guias-ubuntu.ourproject.org/index.php?title=Gu%C3%ADa_Base/Mil_y_un_programas_libres_y_gratuitos/Ofim%C3%A1tica_con_OpenOffice.org)
  51. [www.gnome.org/projects/evince](http://www.gnome.org/projects/evince)
  52. [www.mozilla.org](http://www.mozilla.org)
  53. [www.gimp.org.es](http://www.gimp.org.es)
  54. [www.linuxfocus.org/Castellano/January2002/article132.shtml](http://www.linuxfocus.org/Castellano/January2002/article132.shtml)
  55. [www.irccast.com/lv/categories/viewlist/kl0.htm?curs=7&q=windows&](http://www.irccast.com/lv/categories/viewlist/kl0.htm?curs=7&q=windows&)
  56. [www.dte.us.es/extuniv/linux/avanzado/ejercicios/unidad\\_2-6\\_ejercicios.pdf](http://www.dte.us.es/extuniv/linux/avanzado/ejercicios/unidad_2-6_ejercicios.pdf)
  57. [gparted.sourceforge.net](http://gparted.sourceforge.net)
  58. [www.bacula.org](http://www.bacula.org)
  59. [www.guia-ubuntu.org/index.php?title=K3B](http://www.guia-ubuntu.org/index.php?title=K3B)
  60. [www.virtualbox.org](http://www.virtualbox.org)
  61. [packages.ubuntu.com/feisty/net/gnome-btdownload](http://packages.ubuntu.com/feisty/net/gnome-btdownload)
  62. [filezilla-project.org](http://filezilla-project.org)
  63. [uml.sourceforge.net](http://uml.sourceforge.net)
  64. [www.eclipse.org](http://www.eclipse.org)
  65. <http://www.gambasdoc.org/help/doc/diffvb?es>
  66. [www.clamav.net](http://www.clamav.net)
  67. [www.openssh.org/es](http://www.openssh.org/es)
  68. **nova el doc**
  69. **tesis de michel**
  70. tesis de dayrom
  71. **tesis de yaima**