



Guía de migración de aplicaciones de escritorio y servidores a aplicaciones de software libre, para Organismos Públicos chilenos.

Esta Guía de Migración forma parte del  
Estudio de Uso de Software Libre comisionado por la Estrategia Digital al  
Departamento de Ciencias de la Computación de la  
Pontificia Universidad Católica de Chile.  
Edición 2008-2009

Responsable del Estudio: Jens Hardings Perl, Pontificia Universidad Católica de Chile  
Participantes: Alberto Cerda Silva, Andrés Junge Mac-Evoy, Guillermo Lama Antola,  
Claudio Ruiz Gallardo, Marcos Sepúlveda Fernández, Marco A. Zúñiga Yáñez.

Se autoriza la reproducción total o parcial de los contenidos de este documento, citando la fuente de autoría

## Tabla de Contenidos

1	Introducción.....	5
2	Gestión de la migración.....	6
2.1	Estrategia .....	8
2.2	Migración Total.....	10
2.3	Migración Parcial.....	11
2.4	Metodología de la migración.....	12
2.5	Plan de Trabajo y calendarización.....	12
2.5.1	Decisión.....	12
2.5.2	Planificación.....	14
2.5.3	Recopilación de información de los usuarios.....	14
2.5.4	Ingeniería de detalle del plan de trabajo.....	15
2.5.5	Implantación.....	15
2.5.6	Pruebas .....	16
2.5.7	Integración.....	16
2.5.8	Capacitación para administradores.....	17
2.5.9	Capacitación para usuarios.....	17
2.5.10	Gestión de atención continua y en sus diferentes niveles.....	18
2.6	Aspectos para facilitar un proceso de migración.....	20
2.7	Recomendaciones a futuro.....	21
2.8	Factores de éxito para el proceso de migración.....	21
2.8.1	Aspectos Técnicos.....	21
2.8.2	Aspectos Organizacionales.....	22
3	Guía de Software y detalles técnicos.....	23
3.1	Grupo Aplicativo de Escritorio.....	23
3.1.1	Ofimática.....	23
3.1.2	Correo electrónico .....	27
3.1.3	Agendas y calendarios colaborativos .....	28
3.1.4	Clientes WWW.....	30
3.2	Repositorios de Datos.....	31
3.2.1	Bases de Datos.....	31
3.2.2	Servidor WEB/ Servidor de Aplicaciones.....	34
3.2.3	Servidores de Archivo (File Sharing).....	37
3.2.4	Impresión.....	38
3.2.5	Respaldo.....	39
3.3	Mensajería.....	40
3.3.1	Servicio de Correo.....	40
3.3.2	Servicios de Correo WEB.....	42
3.4	Servicios seguridad .....	42
3.4.1	Identidad electrónica.....	42
3.4.2	Autenticación centralizada.....	43
3.4.3	Antivirus .....	43

3.4.4 VPN.....	44
3.4.5 Firewall.....	45
4 Consideraciones .....	46
5 Glosario.....	47

## 1 Introducción

La migración es un proceso largo que generalmente lo abordamos pensando solo en la parte ingeniería. Como herencia de tiempos anteriores en los cuales los usuarios finales tenían interfaces al usuario simples, sin embargo software y las componentes de hardware eran complejos. Se requerían ingenieros altamente especializados para realizar una migración. El grado en que los usuarios intervenían era mínimo. Solo reportes de inconsistencia de datos o ausencia de funcionalidades eran reportados. Por otro lado, dado que los sistemas informáticos no eran parte de la vida cotidiana de la gente, los procesos de migración eran estrictos y con una documentación rigurosa. La situación ha cambiado, tenemos en nuestros escritorios computadores con más potencia de procesamiento que antes, permitiendo a los usuarios realizar tareas que antes no eran posibles. En los sistemas actuales no existe la rigidez relacionada a la usabilidad. Esto implica que existen diversas formas para realizar la misma tarea. De esta manera hemos acostumbrado a los usuarios a privilegiar maneras individuales para el trabajo y no reglas estrictas para realizar las tareas. Por este motivo, si necesitamos hacer un cambio, se presentará una resistencia natural por parte del usuario a seguir utilizando sus formas de trabajo. La planificación de un cambio guiado para todas las maneras de trabajo sería costosa y poco efectiva lo cual dificultaría un proceso de migración.

Los sistemas y la informática están mucho más cerca del usuario. Los computadores ya no son solamente repositorios de datos específicos ni tampoco de procedimientos documentados. Existe una diversidad de funciones, usos, datos e intercambio de información que ha convertido este tema en un tema complejo y altamente granulado. Sumado a estos cambios el costo de los sistemas informáticos ha disminuido y a llevado a que dejemos ciertos procesos costosos en un análisis simplista, pensado que la automatización ha llegado a un grado de madurez que nos permite ausentarnos de cualquier proceso. Existe entonces el riesgo de olvidar la rigurosidad y las exigencias necesarias para realizar un proceso de migración de forma exitosa.

El presente documento describirá la forma y detallará las condiciones necesarias para facilitar procesos de migración. Las tecnologías detalladas en este documento deben ser revisadas para la utilización de la versión correspondiente, siendo necesario revisar las consideraciones para la versión de software y cada proceso en particular. Las indicaciones detalladas en este documento son recomendaciones generales basadas en la experiencia y documentación de otras instituciones que han puesto énfasis en detallar dichos procesos.

La guía esta separada en dos partes. La primera describirá cómo organizar nuestros recursos desde el punto de vista organizacional. Principalmente los recursos humanos para realizar y gestionar la migración. También las personas que se verán afectadas por la migración. La segunda parte mostrará software y sus posibles caminos de migración y recomendación para esto.

## 2 Gestión de la migración

La etapa más importante en el proceso de migración será la planificación de proceso completo. Definir métodos y mecanismos de control para la migración será esencial para poder tener éxito en nuestra tarea. Generalmente nos concentramos en detalles técnicos, que nos consumirán recursos y nos impedirán lograr un resultado íntegro. Procesos de migración sin documentación ni planificación serán generalmente un desastre. Desde cambiar un motor de base de datos RDBMS hasta un navegador o browser sin un plan correspondiente implicará eventos no controlados. No lograremos realizar cambios, invalidaremos a nuestro equipo interno de trabajo y provocaremos pérdidas no cuantificables en horas de trabajo de toda la organización. Por este motivo revisar todos los puntos detallados a continuación, solo será el comienzo de un plan exitoso. Como en todo proyecto, el riesgo deberá ser controlado. No bastará acudir a un especialista en software o hardware, tampoco a un equipo con conocimientos avanzados sobre las herramientas de software. Será necesario, conocer las necesidades de la organización y también la manera de funcionar.

La gestión de la migración debe ser planificada y deben existir hitos predefinidos. Estos hitos deberán ser controlados en cada etapa revisando el cronograma y precedencia de cada actividad. En cada hito se cumplirán nuevas tareas, para que se logre la funcionalidad deseada en el proceso migración. Es importante tener en cuenta siempre, que la migración es simplemente un reemplazo de las herramientas productivas. Esto implica que las mejoras u otros cambios en la funcionalidad original no son estrictamente parte de la migración. Por este motivo, a priori deberá quedar declarada la funcionalidad que se desea obtener al finalizar el proceso de migración. Por ejemplo, cambiar el sistema de correo no es solamente reemplazar el software. En realidad es proveer a los usuarios finales las misma funcionalidades que actualmente tiene con el servicio de correo que este operando. Debemos tener en cuenta que el concepto de funcionalidad estará ligado a la forma de uso del usuario final. Por este motivo, si la funcionalidad es la misma, pero la forma cambia implicará que será una herramienta potencialmente inútil. Si la funcionalidad es el envío de correo y cambiamos la forma tradicional de estructurar el destinatario por otra, el usuario se verá incapacitado para realizar esta tarea. Desde la llegada, hasta el envío de un correo. Sin embargo, como parte del proceso sería deseable incorporar nuevas funcionalidades colaborativas. Es decir debemos dejar claro que tareas serán parte estricta de la migración y cuales serán nuevas funcionalidades o piezas de software. Estas tareas deberían quedar aparte del plan de migración por los siguientes motivos:

- Involucrará conocimientos nuevos para administradores de sistema y usuarios.
- Probablemente interoperará con otras piezas de software que puedan verse alteradas en funcionamiento y/o rendimiento.
- Se prevé un aumento de riesgo en la tasa de fallas reportadas por los usuarios.
- Aumentará el riesgo para que las funcionalidades o formas de uso originales no se

cumplan

Por lo mencionado anteriormente es recomendable separar los procesos de renovación y migración en el plan de gestión.

Como parte de la gestión debemos contemplar el proceso de elección del software. El análisis y elección del software a utilizar influirá en el tiempo y metodología de la migración. También para procesos de migración en los cuales se consideren actualizaciones, deberemos tener eventualmente considerar nuevas herramientas de apoyo y personal calificado en el software y las herramientas que utilizaremos.

Los aspectos más importantes en la elección del software que utilizaremos son los siguientes:

- Soporte del proveedor (información pública disponible en la red o servicios pagados)
- Desarrollo anexo necesario para que el paquete de software elegido cumpla con las funcionalidades.
- Compatibilidad y eventual interoperabilidad con otros paquetes de software en uso.
- Evaluar las funcionalidades que no estén en el nuevo paquete de software y revisar si podremos prescindir de ellas.

La tarea de la elección de software se verá facilitada en la medida que tengamos más antecedentes sobre el software y las versiones disponibles. No debemos olvidar que cuando revisemos las alternativas de software a utilizar, debemos tener en cuenta la compatibilidad con versiones anteriores. Usualmente tendemos a reemplazar una pieza de software encantados por la nuevas funcionalidades y potencialidades de este, perdiendo el foco en la integración funcional con todos los componentes. También debemos considerar la madurez del software que vamos a seleccionar para la migración.

En la madurez están implícitos aspectos que debemos considerar como:

- Disponibilidad de herramientas de administración
- Tiempo de vida de la última versión
- Tasa de errores o bugs reportados
- Tasa de errores o bugs corregidos
- Madurez de los estándares utilizados por el software
- Popularidad del software, medido en cantidad de usuarios y reportes de éxito

Estos puntos nos ayudarán a disminuir los riesgos ante bugs no corregidos que impliquen un retraso en el proceso de migración.

Para facilitar la gestión de la migración podremos ver cuatro grandes etapas. Cada una de estas etapas posee actividades que podrán ser independizadas para controlar los avances de una migración. En un ambiente ideal, podremos dejar cada proceso casi en forma

independiente y causalmente ordenados. Sin embargo en la práctica, es probable que después de una primera ejecución de cada etapa, debamos ajustar cambios en cada una de estas.

## **1. Configuración**

Comúnmente pensamos que la configuración de un sistema es portable. Lamentablemente ni siquiera podemos asegurar que de una versión a otra del mismo software las configuraciones sean replicables. Por este motivo debemos unir en un formato común las antiguas y nuevas configuraciones. Revisando en cada una las implicancias que tiene sobre el software, como también sobre la funcionalidad y operación de nuestra plataforma.

## **2. Administración**

Concretar la capacidad de gestionar el cambio del sistema de salida y el nuevo sistema. Velar para que nuestro equipo de administración sea capaz de enfrentar la tarea de gestionar y operar el nuevo sistema, desde la transición hasta el reemplazo total. El manejo de los recursos humanos del equipo de administración podrá ser de diversas maneras. Sin embargo no debemos olvidar que este equipo deberá ser capaz de manejar incidentes y también la operación.

## **3. Integración funcional**

Sin duda esta tarea es una de las más complejas. En esta etapa deberemos velar por una integración vertical de los componentes. Esto implicará que debemos velar para que todos los componentes estén conectados, de manera que podamos lograr la funcionalidad que se planificará. No bastará con instalar y dejar operando los paquetes de software. Práctica común en la administración del sistema. Podríamos comparar esto con la construcción de una carretera olvidando la señalética y conexión con otros caminos. Nuestra implantación y migración deberá tener sentido para todos quienes intervengan en la migración.

## **4. Integración de datos**

La integración de los datos a nuestro nuevo escenario será la tarea la parte final de este grupo de etapas. Requerimiento esencial es que nuestros datos almacenados puedan ser nuevamente transformados en información por nuestras nuevas aplicaciones. No olvidemos que en el tiempo las maneras de almacenar datos han cambiado y probablemente hayamos conservado estándares antiguos. La compatibilidad en el almacenamiento será indispensable para poder velar por el funcionamiento. En este caso es bueno destacar, la codificación.

### **2.1 Estrategia**

Paralelamente a la organización de la migración, deberemos en cada etapa visualizar una estrategia adecuada. Por este motivo es necesario conocer ventajas y riesgos de cada una. Con esta información podremos dimensionar de mejor forma nuestros costos y cuantificar los



riesgos que significará el proceso de migración.

Decidir sobre la velocidad del proceso de migración una vez iniciado. Estas son las principales alternativas:

- **Instantáneo o Súbito:** En este proceso el cambio se realizará de un momento a otro. Sin duda que el proceso podría durar horas en su totalidad. Sin embargo la idea es que de un momento a otro el nuevo sistema aparezca en línea para los usuarios. Generalmente este cambio puede ser costoso y poco efectivo. Acumularemos todo los posibles errores en un solo instante de tiempo. Probablemente acumulando reclamos de todos los usuarios simultáneamente. Esto congestionará nuestro sistema de ayuda a los usuarios, provocando la no disponibilidad de los sistemas. Los eventos inesperados mantendrán a todo el equipo encargado de la migración ocupado, impidiendo que este se preocupe de las tareas principales planificadas en la migración. Las desventajas de un cambio repentino de la plataforma implicarán con seguridad una gran cantidad de eventos inesperados y situaciones que se tornarán incontrolables. En resumen, es imposible prever ante una migración todos los inconvenientes que se presentarán. Por este motivo el cambio brusco de una plataforma a otra debe ser evitado bajo cualquier circunstancia. No se recomienda bajo ninguna circunstancia un cambio súbito. De igual manera dejar el sistema antiguo fuera de servicio, con imposibilidad de volver atrás.

Razones para un cambio súbito pueden ser la siguientes:

- Cambios en las regulaciones vigentes que obliguen a una institución reemplazar la plataforma de software para el cumplimiento de nuevas normativas.
  - Por razones técnicas o comerciales debemos descartar el uso de la plataforma actual lo antes posible. Falta de soporte técnico, ausencia de licenciamiento vigente pueden ser alguno de los motivos.
  - Los administradores se verán enfrentados a una sola plataforma posteriormente a la migración. Esto implicará que no tengan que convivir con más plataformas operativas, desgastando los procedimientos de atención y soporte.
  - Enfrentar a usuarios finales y administradores solamente a un cambio. Aunque sea eventualmente más costoso en recursos humanos, el agotamiento de enfrentarse a la nueva solución será una sola vez.
- **Migración gradual en grupos:**

En esta migración podemos dejar el ambiente de salida y el nuevo ambiente conviviendo para grupos de usuarios. De esta manera organizaremos el trabajo en grupos de usuarios y/o de aplicaciones. Los grupos pueden ser de un usuario o más. Los grupos de transición deberán ser organizados para poder realizar una gestión de acuerdo a los recursos que dispongamos para la atención. No olvidemos que ambos métodos deberán ser realizados después de la etapa de pruebas. Sin embargo para la etapa de pruebas puede ser conveniente organizar grupos unitarios con un control

total sobre el proceso de migración. Idealmente clasificaremos a los usuarios con menor resistencia al cambio en los primeros grupos. Ellos proveerán de información valiosa, siendo quienes tendrán mejores posibilidades de gestionar el cambio en forma autónoma. Los recursos, que obviamente serán limitados, serán distribuidos de mejor manera para no dejar grupos de usuarios desatendidos. El grupo final deberá ser el de mayor relevancia en la organización. De esta manera el impacto total será menor.

Fundamentos para tomar la opción gradual son los siguientes:

- Minimizar el impacto económico en instituciones con presupuesto reducido.
- Sistema muy complejos, desde un punto de vista informático, que requieran un cambio gradual para asegurar el funcionamiento continuo.
- Falta de experiencia y conocimientos por parte de los administradores. En el proceso de migración, usuarios y administradores adquirirán el conocimiento necesario que será utilizado.
- En un escenario con usuarios y administradores con alta resistencia al cambio, podremos encontrar mejores métodos para aumentar el grado de adopción al nuevo sistema en forma gradual.
- Estructuras organizacionales complejas.

La forma de abordar la migración dependerá de nuestro objetivo definido cuando decidimos realizar el cambio. En esta fase quedarán definidas las piezas de software que reemplazaremos. Básicamente existirán dos alternativas para realizar la migración definida por el reemplazo total o parcial.

## 2.2 Migración Total

Llamaremos migración total de un sistema, cuando todos los componentes de software son reemplazados sin conservar piezas de software de la plataforma de un conjunto de funcionalidades relacionadas.

Probablemente la más fácil de abordar desde el punto de vista de la infraestructura. En este tipo de migración estaremos enfrentados a reforzar la capacitación a los usuarios. Serán ellos, quienes se enfrenten a un nuevo desafío desde el piloto de la migración hasta el cambio definitivo. Aparentemente la migración total revestirá menores riesgos desde la instalación y proveerá experiencia para el equipo de ingeniería que realice este cambio. Sin embargo, deberemos documentar todas aquellas funcionalidades e interfaces que revistan un cambio mayor para cualquier parte del equipo. La generación de casos de usos para pruebas será fundamental. Con estos casos llegaremos a tener una mayor precisión en el proceso de cambio, siendo útiles para un posterior proceso de capacitación. Este tipo de migración generalmente implica un mayor esfuerzo por todo los grupos involucrados en la

migración. Es importante enfatizar que el proceso de capacitación será imprescindible en todas las etapas y para todos los participantes de la migración y posterior uso de todos los sistemas involucrados.

En este tipo de migración implicará menores riesgos de interoperabilidad. La sustitución completa de una plataforma suele ser más fácil.

### **2.3 Migración Parcial**

Probablemente la más fácil de abordar desde el punto de vista de la infraestructura.

Debemos tener en cuenta un mapa de interoperabilidad claro para el reemplazo parcial de nuestra plataforma. Generalmente en este tipo de cambios sufriremos las consecuencias que nuestros sistemas se verán afectados de una u otra manera en el desempeño funcional. Esto significa que los usuarios, sean administradores o usuarios finales, notaran deficiencias evidentes. Para esto será necesario prever que funcionalidades se verán afectadas en el cambio

Los pasos recomendados a seguir son los siguientes:

- Documentación de la nueva pieza de software
- Mapa de interoperabilidad con los componentes existentes
  - Plataformas propietarias
  - Plataformas desarrolladas a medida
- Configuración de los nuevos componentes
- Configuración de los componentes que no serán reemplazados e interoperarán con la nueva pieza de software
- Laboratorio de pruebas
- Establecer la viabilidad del reemplazo
  - Análisis cuantitativo de las funcionalidades
  - Análisis cualitativo del reemplazo
- Puesta en producción
- Recopilación documentada de los eventos reportados.
  - Fecha
  - Hora
  - Lugar
  - Tipo de usuario
  - Funcionalidad con problemas

- Generar reporte de gestión sobre los eventos.
- Identificar eventos
- Repetir el procedimiento hasta lograr los niveles de satisfacción deseados

## **2.4 Metodología de la migración**

Cualquier ejercicio de migración debe incluir al menos lo siguiente:

1.- Definición y alcances del proyecto:

- a) Nivel de los usuarios, a nivel de usuarios finales y administradores
- b) Tipo de aplicaciones
- c) Topología de red, protocolos de red a utilizar
- d) Condiciones iniciales y estado final deseado después de la migración

Para asegurar que el estado final es realizable deberá ser incorporado al plan una fase de pruebas con un laboratorio de migración donde se realizarán las pruebas de concepto.

Una prueba de conceptos debe incluir lo que se detalle como importante en un ambiente simplificado. El uso de la combinación total de posibilidades de la migración podría convertir el proyecto completo en una tarea de varios meses sin resultados de corto plazo.

## **2.5 Plan de Trabajo y calendarización.**

El plan de trabajo deberá considerar los puntos señalados a continuación. El desarrollo de un plan de trabajo consistente nos permitirá calendarizar todas las actividades para dimensionar en forma eficientes los recursos a utilizar. El tiempo y otros recursos que dispongamos se transformarán en variables que nos entregarán una medida más exacta para el desarrollo de la migración.

### **2.5.1 Decisión**

En el proceso de decisión definiremos qué herramientas serán reemplazadas. Sea en el caso de un reemplazo total o parcial, deberemos considerar desde un punto de vista global todas las etapas en el proceso de decisión. Para facilitar el proceso en la toma de decisión revisaremos la metodología que utilizaremos.

En esta metodología existen cuatro áreas que revisaremos. Estas permitirán abordar la migración desde la gestión necesario y recursos que dispongamos. Además deberemos considerar que productos y estándares seleccionaremos para coordinar todo el proceso de migración.

## **Estrategia**

Revisaremos que tipo de estrategia, de las mencionadas anteriormente puede adecuarse a nuestras necesidades. El reemplazo total, parcial y si este será por partes en un proceso radical o gradual.

## **Plataforma**

Analizaremos las plataformas existentes, y deberán ser detalladas con la mayor cantidad de información. Posteriormente confeccionaremos una nueva lista con las plataformas que deseamos realizar la migración y también con la mayor cantidad de información. La información sugerida, es versión, requerimientos de hardware, requerimientos de software, y funcionalidad deseada. Toda la información que recolectemos servirá para generar una matriz de compatibilidad e interoperabilidad entre las plataformas existentes y las que reemplazaremos. Los elementos de salida, serán aquellos que están operando actualmente. Sistemas operativos, paquetes de software y software propietario. Los elementos de entrada serán aquellos que estén en la planificación para ser instalados o reemplazados como parte de la migración.

La confección de una matriz con los sistemas respectivos para verificar compatibilidad entre las componentes que permanecerán en nuestro ambiente y los componentes de entrada se utilizará para verificar antes de iniciar el proceso de migración el grado de integración que potencialmente se alcanzará. Para esto definiremos elementos o plataforma de salida y elementos de llegadas. Identificando la versión de cada una de ellas y si posee alguna adaptación a medida para nuestra plataforma.

## **Estándares**

Revisar si las plataformas de salida y entrada se adecuan a los estándares documentales y de protocolo para operar. Generalmente en una migración hacia sistemas de código abierto, la utilización de estándares será implícita. Sin embargo al vernos en la obligación de reutilizar código o fuentes de datos propietarias, deberemos tener en cuenta

## **Productos**

Elección de los paquetes de software para el reemplazo y futura migración. En esta elección debemos tener en cuenta la mano de obra calificada a la cual podamos acceder para el desarrollo del proyecto. Las versiones respectivas que utilizaremos. Finalmente buscar experiencias similares apoyadas en asesorías o experiencias de entidades similares.

## **Concepción del proyecto**

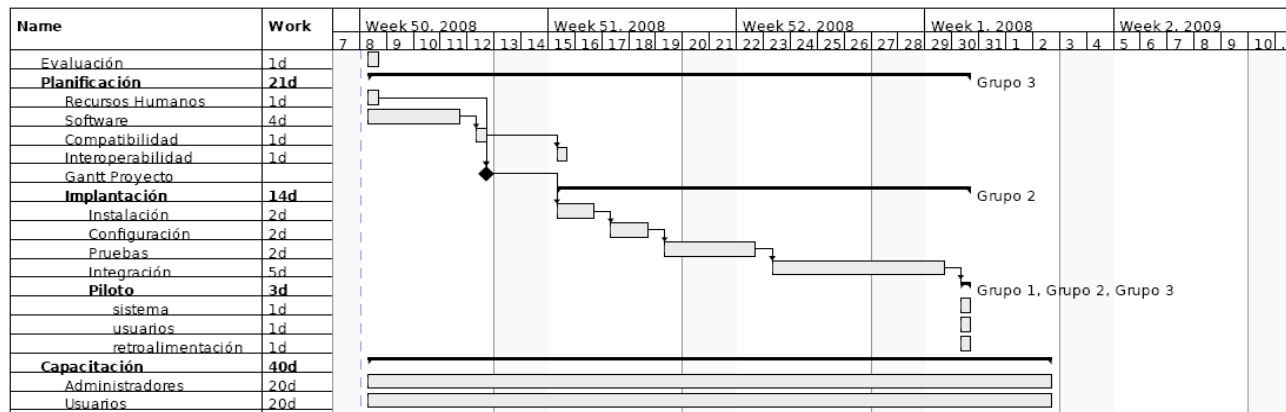
Definición precisa del proyecto, en la cual detallamos los recursos involucrados y dimensionaremos el tiempo necesario para el desarrollo de la migración.

## 2.5.2 Planificación

En la planificación deberemos detallar por escrito todas las etapas que se detallarán a continuación. Con responsables individuales, dimensionamiento de recursos y definición de hitos. Cada tarea deberá ser estructurada, de manera tal que sea explícita la dependencia de otras para poder planificar en el menor tiempo posible la migración.

WBS	Name	Start	Finish	Work	Duration	Slack	Cost	Assigned to
1	Evaluación	Dec 8	Dec 8	1d	1d	19d	0	
2	<b>Planificación</b>	<b>Dec 8</b>	<b>Dec 30</b>	<b>21d</b>	<b>17d</b>	<b>3d</b>	<b>0</b>	Grupo 3
2.1	Recursos Humanos	Dec 8	Dec 8	1d	1d	5d	0	
2.2	Software	Dec 8	Dec 11	4d	4d	1d	0	
2.3	Compatibilidad	Dec 12	Dec 12	1d	1d	1d	0	
2.4	Interoperabilidad	Dec 15	Dec 15	1d	1d	14d	0	
2.5	Gantt Proyecto	Dec 12	Dec 12	N/A	N/A	1d	0	
2.6	<b>Implantación</b>	<b>Dec 15</b>	<b>Dec 30</b>	<b>14d</b>	<b>12d</b>	<b>3d</b>	<b>0</b>	Grupo 2
2.6.1	Instalación	Dec 15	Dec 16	2d	2d	3d	0	
2.6.2	Configuración	Dec 17	Dec 18	2d	2d	1d	0	
2.6.3	Pruebas	Dec 19	Dec 22	2d	2d	3d	0	
2.6.4	Integración	Dec 23	Dec 29	5d	5d	3d	0	
2.6.5	<b>Piloto</b>	<b>Dec 30</b>	<b>Dec 30</b>	<b>3d</b>	<b>1d</b>	<b>3d</b>	<b>0</b>	Grupo 1, Grupo 2, Grupo 3
2.6.5.1	sistema	Dec 30	Dec 30	1d	1d	3d	0	
2.6.5.2	usuarios	Dec 30	Dec 30	1d	1d	3d	0	
2.6.5.3	retroalimentación	Dec 30	Dec 30	1d	1d	3d	0	
3	<b>Capacitación</b>	<b>Dec 8</b>	<b>Jan 2</b>	<b>40d</b>	<b>20d</b>		<b>0</b>	
3.1	Administradores	Dec 8	Jan 2	20d	20d		0	
3.2	Usuarios	Dec 8	Jan 2	20d	20d		0	

Posteriormente utilizando alguna herramienta para manejo de proyectos.



En este ejemplo hemos utilizado planner.

## 2.5.3 Recopilación de información de los usuarios

La información de los usuarios será indispensable en el proceso de migración. Deberemos manejar la información y perfilamiento de cada usuario en forma independiente. Además deberemos garantizar la integridad y privacidad de la información. Idealmente esta tarea

debería consistir solamente en aislar respaldos e imprimir plantillas con perfiles. Lamentablemente lo que ocurre generalmente es la ausencia de los datos necesarios en el proceso de migración.

Definir un repositorio personalizado para los datos de los usuarios. En este repositorio se deberá garantizar la privacidad de los datos locales o centralizados que migraremos.

- Garantizar el almacenamiento de los datos del usuario en los servidores
- Almacenar datos personales de los usuarios. Estos se encuentran generalmente en los equipos de trabajo del usuario. Por este motivo, deberá ser definido un proceso con fechas exactas para el almacenamiento y restauración de los sistemas.
- Verificar políticas e implementaciones de los accesos y permisos de los cuales disponga el usuario.

#### **2.5.4 Ingeniería de detalle del plan de trabajo**

Serán los planos de nuestra migración. Con detalles específicos versión del paquete de software y también directivas de configuración del mismo.

El detalle de la solución deberá ser una directriz del equipo que coordine y maneje la migración. Los documentos resultantes deberán proveer información explícita sobre todo lo instalado y la metodología que se utilizará.

#### **2.5.5 Implantación**

Es un trabajo puramente técnico. Se instalarán y configurarán los sistemas deseados

Antes de la implantación podremos revisar una lista de actividades:

- Revisar el cumplimiento los requerimientos de software, hardware y red
- Conocer protocolo y nomenclaturas del software que implantaremos
- Disponer de privilegios en los sistemas para instalar el software
- Disponer de la mayor cantidad
- Conocer el lenguaje y nomenclatura de los nuevos sistemas. Recomendable será generar un diccionario, en el cual asimilaremos los nuevos conceptos. Además estableceremos las equivalencias frente a posibles diferencias . Esto significa, definir el concepto y detallar el nombre en los sistemas de salida y entrada en la migración.
- Registrar todos los eventos que se presenten en el proceso de implantación.
- Solicitar documentación detallada en el caso que la implantación sea realizada por terceros.

### 2.5.6 Pruebas

La etapa de pruebas es fundamental en el éxito de este proceso. Deberá existir un plan de pruebas documentado. Mientras mayor sea el detalle que logremos, podremos mitigar el riesgo de un fracaso.

Dependiendo de la metodología que utilicemos, podremos involucrar a los usuarios finales o a un grupo de ellos.

Según lo estipulado en las necesidades funcionales, se realizarán las pruebas necesarias. En esta etapa se verificará principalmente:

- Interoperabilidad con el resto de los sistema involucrados
- Funcionalidades requeridas en la formulación del proyecto
- Grado de adaptación de los usuarios

El plan de pruebas deberá considerar como mínimo el registro de los siguientes aspectos:

- Nombre de la prueba
- Objetivo de la prueba. Verificación funcional.
- Nombre del usuario o tester que realizará la prueba
- Fecha y hora
- Software involucrado en la prueba
- Interfaz o mecanismo que será sujeto a la prueba
- Número de la prueba
- Condición de finalización. En caso de no ser exitosa, deberá quedar registrado, indicando las medidas que se adoptarán.

Además deberán quedar registrados todos los errores y ser reportados a la mesa de ayuda o quien realice esta función.

### 2.5.7 Integración

Normalmente un proceso de difícil documentación. Debido a que el número de combinaciones y casos de pruebas suele ser demasiado grande. La implementación de un sistema en forma exitosa, dependerá en gran medida de un alto nivel de integración, en caso de existir otros componentes.

Consolidación del nuevo ambiente de trabajo con todo el software migrado. En esta parte podremos dejar atrás las antiguas piezas de software y comenzar a registrar los evento en producción.



### **2.5.8 Capacitación para administradores**

Las nuevas tareas deberán tener una tabla de equivalencia. Frecuentemente encontraremos solicitudes puntuales, para las cuales se utilizará la jerga del software saliente. En otros casos las tareas específicas que se realizaban en el paquete de software saliente ya no existirán y deberá quedar detallado en las capacitaciones. De esta manera podremos optimizar la adaptación de nuestros administradores y además procurar el uso eficiente del tiempo en la administración de la nueva herramienta.

Un aspecto importante será poder identificar los conocimientos y experiencia que el grupo de administradores tengan o adquieran durante el proceso. Debido a la naturaleza del trabajo de los administradores, ellos tenderán a estar alejados del uso cotidiano de los sistemas. Por este motivo será recomendable que conozcan al menos una parte de la capacitación de los usuarios. Probablemente encontremos resistencia a que los administradores asistan o participen de las capacitaciones. Sin embargo será un factor organizacional que permitirá cohesionar a todo el grupo humano para un mejor resultado de la migración.

Todo el trabajo que realicen terceros en los sistemas deberá incluir un capacitación a los administradores. Es importante que ellos formen parte de todos los trabajos realizados, independientemente que no los implementen, para poder alcanzar un mayor grado de conocimientos. Esto permitirá un menor tiempo de resolución ante eventuales problemas en todos los sistemas.

### **2.5.9 Capacitación para usuarios**

El plan de capacitación deberá ser adaptable a las necesidades de la organización. Recordemos que en un proceso de migración desde el punto de vista de los usuario será un reemplazo. En este reemplazo incorporaremos nuevos elementos para las mismas funcionalidades. Por lo tanto realizar una capacitación en base a las herramientas anteriores puede ser recomendable.

El proceso de capacitación deberá ser claro y estructurado. Esto no implicará que no sea modificable durante el desarrollo de la capacitación. Es fundamental mostrar claridad hacia el resto de la organización de los aspectos que se expondrán.

En el periodo de capacitación, no solo exponremos conocimientos. Será una oportunidad para dar a conocer las labores que implica una migración. Esto permitirá hacer parte a los usuarios de manera integral del proceso de migración. Como mencionamos anteriormente, los sistema actuales dependen en gran medida de los usuarios. Esto significa que si el usuario no entiendo o no se adapta al nuevo sistema, nuestra migración no será exitosa.

Como en todo proceso, muchas veces suponemos el nivel de conocimientos de los usuarios. Por este motivo un plan de capacitación deberá tener las siguientes etapas.

- Evaluación de los usuarios en los sistemas actualmente utilizados
- Construcción de grupos de capacitación, de acuerdo a los niveles evaluados. De esta manera podremos realizar la capacitación en forma más eficiente.
- Calendario de capacitación.
- Detalle de jornadas teóricas y prácticas. Recordemos que si los usuarios no se familiarizan con el sistema, todos los conocimientos no ayudarán al proceso de migración.
- Evaluación final para medir la efectividad de la capacitación.

## 2.5.10 Gestión de atención continua y en sus diferentes niveles

### Atención de usuarios

El proceso de migración necesitará atención a todos los usuarios. Independientemente de la capacitación surgirán dudas o reclamos hacia los nuevos sistema. Por este motivo deberemos asegurar que la atención sea parte integral del proceso de aprendizaje de los usuarios. Nos encontraremos con reclamos inherentes a la migración. Podremos detectar el problema y aplicar una solución. Sin embargo no debemos olvidar que en el proceso de cambio encontraremos reclamos o solicitudes de atención relacionadas solamente al proceso de cambio. Este será un indicador que nos ayudará a medir cual es el nivel de adaptación de los usuarios y administradores al nuevo sistema.

Para la atención de los usuarios recomendamos los siguientes puntos:

- Registro de cada atención
- Detallar el motivo de la atención
- Describir el problema en el registro de atención

### Administración con registro de eventos

Es un punto importante llevar registro de todas las actividades, al igual que los eventos. De esta forma podremos identificar de manera más rápida los eventos y efectuar un cambio de curso eventual en nuestros planes de migración.

### Recursos humanos

Pocos sistemas son migrados en forma exitosa sin considerar a los usuarios finales. Generalmente las migraciones son en sistemas de control automático o procesos incrustados. Pero las áreas de software descritas en esta guía siempre tendrán a una persona en la cadena de implantación y/o utilización.

Entonces debemos pensar que las personas son un aspecto estratégico en la migración.

Identificaremos a los siguientes grupos y mencionaremos ventajas y desventajas para una migración en cada caso.

Tipos de usuarios	Ventaja	Desventaja
Administradores	Usuarios con conocimientos formales	Usuarios autodidactas
Usuarios Finales	Nivel de conocimientos	Necesidades reales para la organización
Soporte	Técnicos capaces de interactuar con los problemas de las personas y no solamente del computador.	Técnicos con ordenes específicas
Desarrolladores	Conocedores de técnicas como extreme programming, que permitan realizar ajustes a medida en forma eficientes	Ajustados a procedimientos y sujetos a normas estrictas.

Dado que es probable que no nos encontremos con el caso ideal de los usuarios que necesitamos por diversos motivos clasificaremos nuestra organización de la siguiente manera a todos los usuarios finales. Todos los aspectos de la planificación y ejecución de la migración deberán contemplar el tiempo de aprendizaje y asimilación del nuevo software. Intuitivamente podremos notar que a mayor cantidad de usuarios, más recursos en nuestras mesas de ayuda y/o personal de atención técnico utilizaremos.

Describiremos tres tipos de usuarios finales que será recomendable identificar:

### **1. Usuarios Avanzados**

Este tipo de usuarios posee conocimientos avanzados en diversas áreas. Normalmente los reconocemos fácilmente en la organización, pero nunca hemos medido sus conocimientos. Esto no nos permitirá ayudarnos de ellos para el proceso en forma medida. Sin embargo podremos contar con ellos para la retroalimentación en la etapa de implantación. Debemos encontrarlos y formar con ellos un grupo de trabajo de avanzada.

Ellos deberán ser los primeros que utilicen el sistema migrado en su etapa de pruebas e implantación.

### **2. Usuarios específicos**

Dedicados usualmente a utilizar herramientas productivas sin familiarizarse con el entorno en forma completa.

### **3. Usuarios básicos**

A ellos los reconoceremos por su lejanía con los sistemas de información en la organización. No podemos identificar cual es el nivel de rechazo con el nuevo sistema, sin embargo deberemos recoger los reclamos, en lo posible en un sistema que permita la clasificación y búsqueda de estos. Deseable será registrar el estado de nuestros reclamos para poder incorporarlos a un proceso de mejora continua.

De la misma manera calificaremos a nuestros administradores de sistema

#### **1. Administradores verticales**

Los podremos identificar porque querrán estar en todas las tareas que estemos realizando. Sin una característica clara, ellos estarán dispuestos a resolver cualquier problema. Una limitante común en este tipo de administradores es que son eficaces pero poco eficientes en tareas de mayor envergadura. La eficiencia decrece a medida que la tarea se vuelve estructurada.

#### **2. Administradores de aplicación**

Se remiten en la mayoría de los casos a conocer una herramienta en particular. Debemos distinguir que un administrador de aplicación puede ser experto en sistemas de correo. Sin embargo esa especialización generalmente esta asociada a un producto en particular. Debemos identificar si el administrador de tarea será para la aplicación saliente o entrante.

#### **3. Administradores de tarea**

Sistemáticos y muy estructurados. Ellos no avanzarán si un plan no esta bien detallado. Ideales para realizar tareas rutinarias que no podrán ser automatizadas.

## **2.6 Aspectos para facilitar un proceso de migración**

Muchas de las aplicaciones que se migrarán, funcionarán en ambientes desconocidos para la mayoría de los usuarios. Entendiendo que lo nuevo siempre puede paralizar a los usuarios frente a lo desconocido, es recomendable fijar en la estrategia una sustitución parcial. La sustitución deberá comenzar por el reemplazo en plataformas conocidas por una herramienta familiar. Un buen ejemplo puede ser la sustitución de cualquier navegador por uno FLOSS. En términos de usabilidad, la experiencia para el usuario será similar. Esto no solo facilitará el proceso de migración en si mismo, también permitirá que los usuarios el nivel de aceptación a las soluciones sea mayor.

Visto desde otro punto de vista, reemplazar el DNS o DHCP por una herramienta FLOSS

puede traer los mismo resultados en otro nivel de usuarios. Frente a herramientas ampliamente probadas, el reemplazo transparente de un sistema, puede ser anunciado como un éxito. Recordemos que generalmente los usuarios y administradores no miden su nivel de satisfacción, sino todo lo contrario su nivel de insatisfacción. Aunque esto pueda parecer simplemente un juego de lógica, en la práctica se traducirá en los siguiente.

- Caso de migración exitoso no anunciado implicará la validación del equipo técnico frente a los cambios y generando confianza y ayuda colectiva de todos los usuarios.
- Caso de migración parcialmente exitoso, podría llevar a saturar nuestro nivel de atención a los usuario producto de supuestos frente a la nueva solución.

Por lo tanto, aunque un proceso de migración sea transparente, es necesario anunciar sobre el proceso y posteriormente no solo recoger reclamos y ayudas, sino que también hacer parte a todo el equipo de usuarios del nuevo logro alcanzado.

## **2.7 Recomendaciones a futuro**

El proceso de migración acabará una vez instaladas las nuevas piezas de software en los lugares planificados. Sin embargo deberemos tener en cuenta los siguientes aspectos a futuro.

- Todos los desarrollos sobre las nuevas plataformas mas deberán estar ajustados a estándares de compatibilidad de las herramientas utilizadas. El caso más común es el desarrollo de herramientas WEB. Para esto los validadores del W3C facilitarán sin duda, la compatibilidad con nuestras nuevas herramientas.
- Fomentar el uso de estándares abiertos para documentos.
- Fomentar el uso de protocolos abiertos para intercambio de información
- Asegurar una metodología de desarrollo que asegure portabilidad, independencia de los datos y la lógica de negocios.
- Evitar los servicios con protocolos propietarios. De esta forma los usuarios se verán obligados a usar los estándares definidos por la administración.
- También debiéramos recomendar la toma de decisión consciente en casos que puedan atarnos a una tecnología (aunque sea FLOSS). Ejemplo: la decisión de usar SQL estándar o el optimizado pero específico a la plataforma utilizada actualmente generalmente no se hace en forma explicita.

## **2.8 Factores de éxito para el proceso de migración**

### **2.8.1 Aspectos Técnicos**

Cada elección de plataforma o software será única debido a las características particulares.

Sin embargo podremos revisar aspectos que no deberemos dejar de lado:

- Documentación de operación y uso del software que se utilizará en la migración
- Verificar la elección de las herramientas a utilizar. Deberemos verificar que la relación de madurez y funcionalidades que nos entregue el software sea útil para nuestro propósito.
- Verificar interoperabilidad y compatibilidad con todo los sistemas instalados. Inclusive con aquellos que no sean parte de la migración. Evitará posibles colisiones y mal funcionamiento de las aplicaciones.
- Identificar las funciones propietarias en los paquetes de software de entrada y salida, para poder identificar un riesgo en el fracaso del proyecto.
- Definición funcional detallada para poder identificar de mejor manera el cumplimiento de todas las necesidades.

### **2.8.2 Aspectos Organizacionales**

Podríamos pensar que una migración es únicamente un trabajo técnico. Sin embargo no debemos olvidar que el aspecto organizacional del equipo de trabajo y los usuarios involucrados será un factor importante en el desarrollo de la migración. El énfasis en este punto, es que usualmente los equipos que lideran una migración son de naturaleza técnica. Por este motivo se hace un mayor esfuerzo en consideraciones de software y hardware, dejando como segunda prioridad el factor humano. Aspectos claves para conducir el proyecto en buena forma son los siguientes:

- Formulación de un proyecto específico con metas concretas
- Formación del equipo de profesional en forma estructurada
- Validación en la organización del equipo que liderará la migración
- Distinguir al equipo de trabajo del equipo directivo que planificará y tomará decisiones
- Definición de responsables y competencias de los involucrados en el proceso
- Entrenamiento apropiado en caso de no poseer la competencias necesarias
- Mostrar el plan de trabajo, para que todos los integrantes entiendan que forman parte de un solo proceso. Usualmente puede que parte del equipo no entienda que son parte de un proceso integral, por lo cual se dificultará el trabajo de integración.

## **3 Guía de Software y detalles técnicos**

### **3.1 Grupo Aplicativo de Escritorio**

#### **3.1.1 Ofimática**

##### **Tecnologías**

Actualmente las tecnologías disponibles son alternativas que pretenden reemplazar el conjunto Office de Microsoft™ (MS-Office). El intento por independizarse de la plataforma MS-Office ha resultado al menos difícil, debido a la gran cantidad de personas que utilizan esta herramienta en dos aspectos. El primero es que la gran cantidad de instalaciones de este producto nos obligue a intercambiar en forma periódica archivos con otras instituciones que utilicen MS-Office. El segundo punto, y no menos importante, es que la mayoría de los usuarios que aprenden algún paquete informático, de manera formal o como autodidacta lo hacen utilizando MS-Office. Por este motivo se recomienda que si la migración se realiza desde MS-Office o Lotus sea a los siguientes productos.

Existen también paquetes como Koffice o Gnomeoffice que son una versión reducida de lo que actualmente entendemos como una framework para ofimática.

En cualquier otro caso, la migración no es vital para el funcionamiento, si este es cerrado y acotado a la institución correspondiente.

El grupo de aplicaciones con mayor éxito en compatibilidad es OpenOffice junto a StarOffice. Si bien OpenOffice está originalmente basado en StarOffice, provee mayor cantidad de funcionalidades y soporte de idiomas.

##### **Migración**

No requiere mayor cantidad de trabajo, debido a que la suite OpenOffice reconoce casi todos los formatos de MSOffice. PowerPoint (.ppt), Excel (.xls) y Word (.doc) sin problemas en general.

El soporte para archivo con incrustaciones OLE (Object Linking and Embedding) es limitado. Por este motivo es recomendable identificar previamente si tenemos aplicaciones institucionales incrustadas en nuestros archivos de datos.

##### **Recomendaciones**

Registrar toda incompatibilidad en un archivo. De esta manera podremos saber que archivos nos provocarán problemas, pudiendo atender de mejor manera los requerimientos.

El uso de macros deberá ser evitado o al menos advertido como riesgoso. El riesgo será no poder contar con las funcionalidades originales del archivo.

La principal amenaza a una migración exitosa, es la sensación de los usuario respecto a

poder utilizar todos los archivos existentes. La base de archivos potenciales, es en la mayoría de los casos.

Resumen compacto de los tipos de archivos soportados para OpenOffice 2.4.

Writer	ABRIR	GUARDAR
	OpenDocument Text, Text Template (.odt, .ott)	OpenDocument Text, Text Template (.odt, .ott)
	OpenOffice.org 1.0 Text Document, Text Template (.sxw, .stw)	OpenOffice.org 1.0 Text Document, Text Template (.sxw, .stw)
	Microsoft Word 2007 document [MS OpenXML] (.docx)*	Microsoft Word 2007 document [MS OpenXML] (.docx)*
	Microsoft Word 97/98, 2000/2001/v.X, XP/2004 document, template (.doc, .dot)	Microsoft Word 97/98, 2000/2001/v.X, XP/2004 document, template (.doc, .dot)
	Microsoft Word 6/95 document, template (.doc, .dot)	Microsoft Word 6/95 document, template (.doc, .dot)
	Microsoft Word para Windows 5 document (.doc)	
	Rich Text Format (.rtf)	Rich Text Format (.rtf)
	StarWriter 3.0-5.0 document, template (.sdw, .vor)	StarWriter 5.0 document, template (.sdw, .vor) StarWriter 4.0 document, template (.sdw, .vor) StarWriter 3.0 document, template (.sdw, .vor)
	StarWriter 1.0, 2.0 document (.sdw)	
	StarWriter DOS document (.txt)	
	AportisDoc (Palm) (.pdb)	AportisDoc (Palm) (.pdb)
	DocBook (.xml)	DocBook (.xml)
	Microsoft Word 2003 XML (.xml)	Microsoft Word 2003 XML (.xml)
	Pocket Word (.psw)	Pocket Word (.psw)
	WordPerfect 4.2 DOS (.wp) WordPerfect 5.1 DOS/Win (.wp) WordPerfect 6.0-12.0 DOS/Win (.wpd) WordPerfect para Macintosh 3.5e (WPD4) WordPerfect para Macintosh 3.0-3.5.4 (WPD3) WordPerfect para Macintosh 2.1 (WPD2) WordPerfect para Macintosh 2.0.x (WPD2) WordPerfect para Macintosh 1.0.x (WPD0)	
	Microsoft Works word processing documents (.wps)	
	Ichitaro 8/9/10/11 document, template (.jtd, .jtt)	
	Hangul WP 97 (.hwp)	
	WPS 2000/Office 1.0 (.wps)	



	Plain Text, Text Encoded (.txt)	Plain Text, Text Encoded (.txt)
	Hypertext Markup Language (.html, .htm)	Hypertext Markup Language (.html)
		Portable Document Format (.pdf) (export)
		Extensible HyperText Markup Language (.xhtml) (export)
<b>Calc</b>	<b>ABRIR</b>	<b>GUARDAR</b>
	OpenDocument Spreadsheet, Spreadsheet Template (.ods, .ots)	OpenDocument Spreadsheet, Spreadsheet Template (.ods, .ots)
	OpenOffice.org 1.0 Spreadsheet, Spreadsheet Template (.sxc, .stc)	OpenOffice.org 1.0 Spreadsheet, Spreadsheet Template (.sxc, .stc)
	Microsoft Excel 2007 document [MS OpenXML] (.xlsx)*	Microsoft Excel 2007 document [MS OpenXML] (.xlsx)*
	Microsoft Excel 97/98, 2000/2001/v.X, XP/2004 spreadsheet, template (.xls, .xlw; .xlt)	Microsoft Excel 97/98, 2000/2001/v.X, XP/2004 spreadsheet, template (.xls, .xlw; .xlt)
	Microsoft Excel 5/95 spreadsheet, template (.xls, .xlw; .xlt)	Microsoft Excel 5/95 spreadsheet, template (.xls, .xlw; .xlt)
	Microsoft Excel 4.x Win spreadsheet, template (.xls, .xlw; .xlt)	
	Data Interchange Format (.dif)	Data Interchange Format (.dif)
	dBASE (.dbf)	dBASE (.dbf)
	Rich Text Format (.rtf)	
	StarCalc 3.0 - 5.0 spreadsheet, template (.sdc, .vor)	StarCalc 5.0 spreadsheet, template (.sdc, .vor) StarCalc 4.0 spreadsheet, template (.sdc, .vor) StarCalc 3.0 spreadsheet, template (.sdc, .vor)
	StarCalc 1.0 spreadsheet (.sdc)	
	Lotus 1-2-3 1.x, 2.x spreadsheet (.wk1, .wks)	
	SYLK (.slk)	SYLK (.slk)
	Comma Separated Values (.csv, .txt)	Comma Separated Values (.csv)
	Microsoft Excel 2003 XML (.xml)	Microsoft Excel 2003 XML (.xml)
	Pocket Excel (.pxl)	Pocket Excel (.pxl)
	Quattro Pro 6.0 (.wb2)	
	Hypertext Markup Language (.html, .htm)	Hypertext Markup Language (.html)
	Web Page Query (.html)	
		Portable Document Format (.pdf) (export)
		Extensible HyperText Markup Language (.xhtml) (export)
<b>Impress</b>	<b>ABRIR</b>	<b>GUARDAR</b>
	OpenDocument Presentation, Presentation Template (.odp, .otp)	OpenDocument Presentation, Presentation Template (.odp, .otp)
	OpenOffice.org 1.0 Presentation, Presentation Template (.sxi, .sti)	OpenOffice.org 1.0 Presentation, Presentation Template (.sxi, .sti)
	Microsoft PowerPoint 2007 document [MS	Microsoft PowerPoint 2007 document [MS

	OpenXML] (.pptx)*	OpenXML] (.pptx)*
	Microsoft PowerPoint 97/98, 2000/2001/v.X, XP/2004 presentation, template (.ppt, .pps; .pot)	Microsoft PowerPoint 97/98, 2000/2001/v.X, XP/2004 presentation, template (.ppt, .pps; .pot)
	OpenDocument Drawing (.odg)	OpenDocument Drawing (.odg)
	OpenOffice.org 1.0 Drawing (.sxd)	OpenOffice.org 1.0 Drawing (.sxd)
	StarDraw 3.0/5.0 drawing (StarImpress) (.sda, .sdd)	StarDraw 5.0 drawing (.sda) StarDraw 3.0 drawing (.sdd)
	StarImpress 4.0/5.0 presentation, template (.sdd, .sdp; .vor)	StarImpress 5.0 presentation, template (.sdd, .vor) StarImpress 4.0 presentation, template (.sdd, .vor)
	Computer Graphics Metafile (.cgm)	
		Hypertext Markup Language (.html) (export)
		Portable Document Format (.pdf) (export)
		Shockwave Flash (.swf) (export)
		Extensible HyperText Markup Language (.xhtml) (export)
		Windows Bitmap (.bmp) (export)
		Enhanced Metafile (.emf) (export)
		Encapsulated PostScript (.eps) (export)
		Graphics Interchange Format (.gif) (export)
		Joint Photographic Experts Group (.jpg) (export)
		OS/2 Metafile (.met) (export)
		Portable Bitmap (.pbm) (export)
		Macintosh PICT (.pct) (export)
		Portable Graymap (.pgm) (export)
		Portable Network Graphics (.png) (export)
		Portable Pixmap (.ppm) (export)
		PlaceWare Presentation Slide Set (.pwp) (export)
		Sun Raster Image (.ras) (export)
		Scalable Vector Graphics (.svg) (export)
		StarView Metafile (.svm) (export)
		Tagged Image File Format (.tif) (export)
		Windows Metafile (.wmf) (export)
		X Pixmap (.xpm) (export)
<b>Draw</b>	<b>ABRIR</b>	<b>GUARDAR</b>
	OpenDocument Drawing, Drawing Template (.odg, .otg)	OpenDocument Drawing, Drawing Template (.odg, .otg)
	OpenOffice.org 1.0 Drawing, Drawing Template (.sxd, .std)	OpenOffice.org 1.0 Drawing, Drawing Template (.sxd, .std)

	StarDraw 3.0/5.0 drawing, template (.sda, .sdd; .vor)	StarDraw 5.0 drawing, template (.sda, .vor) StarDraw 3.0 drawing, template (.sdd, .vor)
	StarDraw 2.0 (.sgv)	
	StarWriter Graphics Format (.sgf)	
		Hypertext Markup Language (.html) (export)
		Portable Document Format (.pdf) (export)
		Shockwave Flash (.swf) (export)
		Extensible HyperText Markup Language (.xhtml) (export)

### 3.1.2 Correo electrónico

La mayoría de los clientes de correo electrónico posee una interfaz comprensible para el usuario. Esto será fundamental para que la nueva herramienta de correo electrónico sea exitosa.

Los aspectos generales para la migración de una nueva herramienta de correo final serán:

- Protocolo de intercomunicación entre el cliente y el servidor de correo
- Eficiencia en el manejo de archivos

#### Tecnologías

- **Evolution**

Evolution o Novell Evolution es un gestor libre de información personal y de trabajo en grupo para GNOME, desarrollado originalmente por Ximian ahora es parte oficial del escritorio de GNOME. Combina administración de correo electrónico, calendario, agenda y lista de tareas. Forma parte del conjunto GNOME Office.

- Conectividad integrada con Novell GroupWise
- Conectividad integrada con Microsoft Exchange
- Manejo de cuentas de correo mediante IMAP
- Capacidad para S/MIME, mejora de administración de contactos

- **Thunderbird**

Thunderbird es un cliente de correo electrónico de Mozilla. Es multiplataforma y utiliza el lenguaje de interfaz XUL. Thunderbird soporta IMAP/POP, correo HTML, noticias, RSS, etiquetas, corrector ortográfico incorporado, soporte de extensiones y perfiles para despliegue, buscadores, cifrado PGP, un filtro bayesiano de correo no deseado SPAM y otras funcionalidades. Su principal problema contra rivales como Evolution (sobre plataformas Linux) o, en MS-Windows, está en que adolece de un calendario tan poderoso como Evolution.

- **Kmail**

Kmail es el cliente de correo de la plataforma KDE. Posee soporte para integrar antispam y también soporte criptográfico. La integración para manejo de contactos y otras tareas se produce en forma natural con el ambiente KDE.

### **Migración**

La migración de los clientes de correo electrónico puede ser larga y alta en demanda de recursos. Para poder migrar deberemos revisar las siguientes tareas.

- Verificar los datos almacenados localmente.
- Asegurar el traspaso de información al nuevo cliente de correo
- Verificar el protocolo de conexión al servidor IMAP, POP3 u otro
- Incluir en el sistema de respaldo de la organización los nuevos archivos.

Es probable que dispongamos de herramientas que nos ayuden a importar los datos, sin embargo deberemos verificar la integridad de la información rescatada en el nuevo cliente.

### **Recomendaciones**

Debido a que el servicio de correo funciona en forma continua, es recomendable tomar las siguientes precauciones.

- Verificar que no se interrumpa la recepción de correo
- Habilitar una pasarela para garantizar el paso de correo y almacenamiento en forma transitoria
- Almacenar el registro de logs para evitar pérdidas de correo o distorsiones en la entrega.

### **3.1.3 Agendas y calendarios colaborativos**

Las agendas o calendarios colaborativos no han alcanzado la madurez necesaria en un paquete de software abierto. La ausencia de normas de comunicación y estándares colaborativos ha producido un efecto negativo. El resultado es herramientas inmaduras que han privilegiado ciertos ámbitos de una agenda tradicional por sobre la conectividad hacia diferentes plataformas de los componentes como el correo o la agenda. Además no se ha alcanzado el nivel de usabilidad que posee Outlook con Exchange de Microsoft.

### **Tecnologías**

Debido a que la implementación de un calendario colaborativo no existe en forma monolítica, mencionaremos una integración que se acerca bastante a una solución integral.

- **Zimbra**

Zimbra <http://www.zimbra.com> provee un paquete basado en código abierto que brinda funcionalidades basadas en diversos paquetes de software FLOSS incluidos en la plataforma.

- Postfix
- MySQL
- OpenLDAP
- Apache Tomcat o Jetty
- Lucene
- Verity
- ClamAV
- SpamAssassin
- AMaViS and Amavisd-new
- DSPAM
- Aspell
- James
- Sieve

### **Migración**

La migración de un groupware en base a herramientas de software de código abierto son casi imposibles sin considerar un desarrollo anexo. Básicamente las recomendaciones para adoptar un paquete de software como Zimbra se remiten al estudio de la documentación existente y la utilización de los asistentes automatizados para la migración de datos

El proceso de migración a una agenda colaborativa requerirá experiencia en diferentes ámbitos y recursos de programación en diferentes lenguajes.

### **Recomendaciones**

Si bien, las herramientas colaborativas como Lotus Notes o Exchange posee una gran cantidad de funcionalidades, es necesario definir cual de ellas usaremos en nuestra plataforma. Esto permitirá restringir el ámbito de uso de nuestro groupware, haciendo más fácil la elección de la herramienta de software a utilizar.

Se recomienda revisar los foros y documentación exhaustivamente para poder llegar a un buen resultado. La incorporación de un laboratorio de pruebas será parte esencial para una migración compleja.

### 3.1.4 Clientes WWW

La migración de las herramientas cliente para WEB, comúnmente llamadas navegadores residen siempre en el lado del cliente. Por este motivo debemos tener en cuenta el nivel de nuestros usuarios finales, evaluando su nivel de comprensión y precisión en la detección del problema para no generar reclamos excesivos y sobrecargar las mesas de ayuda.

#### **Tecnologías**

- **Firefox**

El browser OS mas utilizado actualmente. Provee una funcionalidad amplia y resulta ser de fácil instalación y actualización.

Basado en el motor de navegación Gecko.

Navegadores relacionados a Firefox son Netscape y Mozilla, conectados por funcionalidades comunes, compatibilidad de conectores (plug-in)

- **Chrome**

Navegador de la empresa Google. La introducción de este navegador es reciente a la generación del presente documento. Por este motivo recopilar experiencias de usuario sobre esta plataforma ha sido difícil. A diferencia de los otros navegadores, no ha sido posible recoger la experiencia de otros usuarios en la implantación masiva de este navegador. Sin embargo se espera un desarrollo más fuerte entendiéndose que ha tomado un porcentaje importante en la red.

#### **Migración**

Idealmente desearíamos una instalación desatendida en clientes. Sin embargo los clientes WWW son administrados por software de inventario de software en el mejor de los casos.

En primera instancia se deberá instruir a los usuarios sobre posibles incompatibilidades de navegación para ciertos sitios. A priori, este factor no será controlable. Sin embargo para sitios que se encuentren dentro de la institución, que cumplan con la norma W3C será suficiente para asegurar su despliegue y usabilidad.

Revisar la compatibilidad de cada conector (plug-in). Sin duda se debe destacar el conector de Adobe Flash. Este es el conector más utilizado.

#### **Recomendaciones.**

El respaldo permanente de la base de datos local de cada navegador deberá ser respaldada según la política de respaldo para datos de usuarios. El motivo es poder rescatar bookmarks u otros datos personales si el usuario por error de uso en el nuevo navegador no lograra utilizar la herramienta nueva o borrarla de la base de datos.

Los estándares W3C no son respetados por los desarrolladores en muchos casos. Por este motivo es recomendable realizar un estudio de tráfico y detectar los principales sitios que son utilizados, para corregir y/o avisar a los usuarios de las posibles deficiencias.

## 3.2 Repositorios de Datos

### 3.2.1 Bases de Datos

#### Tecnologías

- **PostgreSQL**

Es la base de datos más completa disponible dentro del software de código abierto. Está incluida en la mayoría de las distribuciones de Linux y disponible para diversos sistemas operativos. Incluye características técnicas de bases de datos avanzadas como Write ahead log y Multiversion concurrency control, permiten retornar a un estado de tiempo específico y realizar lectura y escritura independientemente para usuarios concurrentes. Existen también administradores de fácil instalación como Pgadmin y phpPgAdmin.

Posee conectores para ODBC de Microsoft y JDBC para Java. Permitiendo de esta manera acceder a la base de datos desde diferentes aplicaciones y plataformas de desarrollo.

- **MySQL**

Existen dos versiones de MySQL. La primera es MySQL Community edition. Esta versión se distribuye como cualquier software con licencia GNU Public License. Es decir, uno puede acceder al código fuente sin restricciones y por este motivo la clasificamos como software de código abierto. La segunda es MySQL Enterprise edition. A diferencia de la primera esta versión tiene una licencia de uso pagado.

Conocida por su rapidez en el acceso de estructuras de datos simples, ha sido utilizada principalmente en combinación con PHP y Apache sobre Linux. Una gran cantidad de instalaciones de Blogs, sitios livianos de comercio electrónico y sitios dinámicos han sido desarrollados sobre esta base de datos. Posee características como motores de almacenamiento independientes (MyISAM para lecturas rápidas, InnoDB para transacciones e integridad referencial), opción para configuración en cluster, soporte consultas anidadas a partir de la versión 5 y conformidad a las reglas ACID usando los motores InnoDB, BDB y Cluster.

#### Migración

Probablemente uno de los tópicos más relevantes en una migración. Una base de datos

proporciona pocas funcionalidades en sí misma. La mayoría de las veces es accedida desde otra aplicación. Por este motivo debemos identificar los siguientes puntos:

- Datos Almacenados
- Procedimientos almacenados
- Scripts de acceso
- Consultas ANSI

Cada uno de estos puntos puede ser relevante al momento de utilizar la base de datos. Normalmente el punto central de la migración es el traspaso de los datos desde la base de datos de origen hacia la de destino. En este proceso encontraremos diferencias en el nombre de los tipos de los campos. Por ejemplo:

VARCHAR – TEXT

En este caso será posible encontrar equivalencia de tipos

Sin embargo muchas veces dependerá no solamente del tipo de dato a utilizar, sino de la configuración del motor de base de datos para el juego de codificación de caracteres o de la implementación. En el caso que dependiera de la implementación, habrá que revisar como se programó la aplicación verificando el largo máximo.

### **Recomendaciones**

Revisar las funcionalidades avanzadas en cada caso de migración es fundamental. Revisar la lista que se detalla a continuación es necesario para enfrentar cualquier desarrollo posterior con el fin de asegurar la continuidad en la operación.

- Tipos de datos implementados en la base de datos
- Funcionalidades de seguridad
- Funcionalidades orientadas a la disponibilidad como Clustering y Replicación
- Sistemas operativos en los cuales las funcionalidades sigan estando presente
- Licenciamiento para cada una de las plataformas
- Formatos de almacenamiento soportados
- Herramientas administrativas disponibles.
- Herramientas de apoyo y/o complemento disponibles
- Drivers y conectores disponibles para la nueva plataforma
- Dialectos y estándares SQL soportados
- Limitaciones del sistema, como tamaño de los campos o de la misma base de datos
- Medición del rendimiento esperado en la nueva plataforma



Limitaciones inherentes al motor de base datos en uso y su configuración inicial pueden ser los siguientes:

- Juego de caracteres (Encoding)

Aspecto importante que debe ser verificado. Podremos encontrar el caso en el cual el despliegue al usuario parezca coincidente, debido a que terminales y páginas web traducen la codificación. Por este motivo deberemos usar una aplicación para la migración basada en conectores que puedan traducir la codificación. Generalmente los conectores JDBC pueden hacer esta tarea en forma simple.

- Procedimientos de acceso (internos o externos)

En los cambios de versión y/o conector, pueden verse afectadas las rutinas de acceso a la base de datos. Por este motivo será fundamental la prueba de todas las rutinas de acceso y verificar su funcionamiento.

- Funcionalidades de los procedimientos almacenados y triggers

Usualmente existen en casi todos los motores de base de datos. Sin embargo será necesaria una recodificación del procedimiento almacenado o trigger validando la rutinas.

En general una buena recomendación es revisar si la versión de la base de datos a utilizar cumple reglas ACID. Estas reglas consisten en

- Atomicidad: es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.
- Consistencia: es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.
- Aislamiento: es la propiedad que asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información nunca generará ningún tipo de error.
- Durabilidad: es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

El punto más relevante será identificar los aspectos propietarios de la base de datos de origen. En este aspecto deberemos considerar todos los aspectos mencionados en la migración. Desde procedimientos almacenados, hasta los tipos de datos. En cada uno podremos encontrar desde sentencias hasta tipos propietarios que nos impidan realizar una migración sin tener que recodificar o generar una nueva estructura. Por lo tanto, es posible que no solo tengamos que modificar el código o estructura en la base de datos, también posiblemente conectores, drivers e incluso código de nuestra ampliación que accede a la base de datos. Finalmente la identificación y resolución para un parte o pieza propietaria

podría implicar cambios mayores que no estuvieran contenidos inicialmente en la migración de la base de datos.

Para poder entregar recomendaciones puntuales deberíamos saber la base de datos de origen. En este sentido se hace casi imposible entregar recomendaciones puntuales. Sin embargo en un proceso de migración deberemos tener en cuenta los siguientes puntos:

- Estructura de datos y datos almacenados

En la mayoría de los casos podremos encontrar scripts o desarrollar programas que nos ayuden a migrar la estructura y los datos contenidos en la base de datos que estamos utilizando actualmente. Sin embargo, ese será solo el primer paso para lograr que el nuevo motor de base de datos trabaje en forma adecuada.

- Procedimientos almacenados, triggers u otros elementos contenidos en el motor de la base de datos de origen

Usualmente es posible encontrar parte de la lógica de negocios utilizada en el modelo 3-capas en el motor de base de datos. Por lo cual no será conveniente analizar estas estructuras y código aisladamente de las aplicaciones que acceden las base de datos.

- Aplicaciones que acceden la base de datos

Si las aplicaciones se apoyan en mecanismos propietarios o únicos de la base de datos, es posible que nos veamos en la necesidad de recodificar. Por este motivo saber si podemos acceder a una eventual modificación del código será imprescindible para realizar una tarea de migración exitosa.

- Mecanismos de acceso a la base de datos

La mayoría de los conectores o drivers para acceder la base de datos prometen las mismas funcionalidades. Sin embargo, los mecanismos implementados en estos conectores suelen tener relación en el desempeño y eventualmente en los resultados del acceso a la base de datos.

- Procedimientos de administración de la base de datos.

Poseer una DBA (Data base administrator)

En resumen podemos decir que deberemos generar un diagrama de todos los componentes relacionados con la base de datos para poder identificar si alguno de ellos no es posible modificarlo y de esta manera impedir nuestro proceso de migración.

### **3.2.2 Servidor WEB/ Servidor de Aplicaciones**

#### **Tecnologías**

Originalmente los servidores web fueron diseñados para compartir información en texto e imágenes. Con un lenguaje definido HTML (HyperText Markup Language). Actualmente los

servidores web, tienen un número mayor de funcionalidades y pueden brindar servicios que van desde atender a un cliente WWW browser o navegador, hasta ser el pilar de un Web Service WS, para atender a otros programas o piezas de código en requerimientos necesarios para procesamiento y/o despliegue de datos. Sin embargo, el uso de servidores web

- **Apache**

Es el servidor web más utilizado en el Internet. Este servidor web es posible utilizarlo en diversas plataformas y la mayoría de las veces lo encontraremos como parte integral de las distribuciones Linux. Posee una arquitectura , la cual permite integrar diversos módulos

Módulo	Funcionalidad
mod_ssl	Comunicaciones Seguras vía TLS.
mod_rewrite	reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a los navegantes o a los motores de búsqueda en cuanto a como fueron desarrolladas estas páginas).
mod_dav	Soporte del protocolo WebDAV
mod_deflate	Compresión transparente con el algoritmo deflate del contenido enviado al cliente
mod_auth_ldap	Permite autenticar usuarios contra un servidor LDAP.
mod_proxy_ajp	Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets y JSP).
mod_perl	Páginas dinámicas en Perl.
mod_php	Páginas dinámicas en PHP
mod_python	Páginas dinámicas en Python
mod_rexx	Páginas dinámicas en REXX y Object REXX
mod_ruby	Páginas dinámicas en Ruby

mod_aspdotnet	Páginas dinámicas en .NET de Microsoft (Módulo retirado).
mod_mono	Páginas dinámicas en Mono
mod_security	Filtrado a nivel de aplicación, para seguridad

También existen módulos que tienen licenciamiento pagado, para integración con diversas plataformas web propietarias.

Aplicaciones de integración comunes son el módulo para redireccionar los requerimientos JSP (Java server pages) a un contenedor como Tomcat.

La integración más popular es LAMP. Acrónimo para Linux, Apache, MySQL, PHP. Conjunto con el cual, desde el sistema operativo, hasta el lenguaje de programación, es posible servir contenido estático y dinámico.

## Migración

- **Contenido Estático**

La migración de contenido estático es la más simple. No necesitaremos modificar el contenido de los archivos expuestos. Estos generalmente serán archivos de texto con o sin contenido HTML y adicionalmente archivos de imágenes u otros formatos que no necesariamente puedan ser visualizados en un navegador.

Solamente deberemos configurar la extensión de los archivos. Además verificar que en nuestro servidor donde se aloja el servidor web y el sistema operativo de este no presenten problemas al distinguir mayúsculas y minúsculas.

- **Contenido Dinámico**

El contenido dinámico es realmente código que generalmente está en lenguajes interpretados. Este código si es propietario deberá ser reescrito en el nuevo lenguaje. Por ejemplo si tenemos un servidor ASP (Active Server Pages) de Microsoft y migramos a una instalación de Apache y PHP no quedará otra alternativa que escribir nuevamente todo el código.

Otra alternativa será que nuestros programas se encuentren en PHP y solo debemos tomar en cuenta, la conectividad hacia bases de datos y también permisos de ejecución e invocaciones a nombres de archivo en el código

## Consideraciones

Debemos recordar que los contenedores de aplicaciones como tomcat y otros similares no están descritos en detalle en esta guía. Sin embargo es posible tener una interfaz desde apache hacia ellos con algunos conectores.

Dado que los servidores web no solamente atienden usuarios, debemos tomar en cuenta los WS (Web Services) y la integración a otras plataformas,

Tanto el entorno de trabajo de .NET como J2EE proporcionan una plataforma integrada para el desarrollo de servicios web. Lamentablemente en la mayoría de los casos una migración de sistemas propietarios como .NET implicarán recodificación mediante un equipo de desarrollo.

Esta recodificación será más fácil si el desarrollo anterior fue realizado utilizando una arquitectura de 3 capas (3-tier).

### 3.2.3 Servidores de Archivo (File Sharing)

Los servidores de archivos en red permiten a los computadores conectados a una red acceder archivos en forma remota. El almacenamiento es aparentemente local, pero se realiza en el servidor.

#### Tecnologías

- **NFS**

Este es el sistema utilizado en la mayoría de las redes Unix y ha estado en uso durante muchos años. Normalmente no es implementado con directivas de seguridad apropiadas, aunque hay una variante segura que se ha puesto en marcha en algunas variantes comerciales de Unix. NFS consiste en un servidor que exporta archivos desde el computador servidores hacia los clientes de una red IP.

- **OpenAFS**

Es una implementación abierta de AFS (Andrew File System). Su principal característica es que se trata de un sistema de archivos distribuido. <http://www.openafs.org/>

- **Samba**

Samba es un producto que pone en marcha el protocolo SMB de Microsoft. Es fundamental para la integración de los sistemas basados en Windows y el FLOSS, y viene en la mayoría de las distribuciones estándar.

- **Netatalk**

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

## **Migración**

La instalación de un nuevo sistema de archivos será compleja únicamente en la configuración de este. El nuevo sistema de archivos deberá cumplir con los siguientes requisitos como regla general.

- Compatibilidad con los clientes
- Manejo de permisos Lectura, escritura, modificación y ejecución, dependiendo de las características del sistema de salida
- Manejo de usuarios

## **Recomendaciones**

Realizar el respaldo en otro medio del total de los archivos. Con esta medida evitaremos pérdidas involuntarias de información.

Revisar permisos de lectura y escritura en nuestro nueva sistema, y naturalmente la interoperabilidad con los diferentes clientes conectados en nuestra red.

### **3.2.4 Impresión**

#### **Tecnologías**

LPRng (<http://www.lprng.com/>) es una implementación desarrollada activamente del antiguo sistema lpr/lpd de estándar BSD. Contiene un cierto número de mejoras que pueden hacerlo mucho más robusto y fácil de gestionar que los productos originales. LPRng se caracteriza por ser seguro. Sin embargo CUPS ha tomado mayor fuerza por su versatilidad y facilidad de instalación.

CUPS (<http://www.cups.org/>) está pensado como sistema de impresión Unix diseñado para ambientes corporativos. Se basa en el Protocolo de Impresión Internet estándar IPP, e incorpora una función para browsing que permite la distribución automática por la red de los detalles de los nombres y características de las impresoras disponibles. CUPS incorpora también una interfaz de usuario WEB para administrar y configurar las impresoras. Hay programas controladores disponibles para la mayoría de las impresoras.

Kprint y GnomePrint

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

## **Migración**

La migración consistirá en habilitar el sistema y dejarlo en producción. Además de considerar los parámetros necesarios para la configuración, deberemos revisar los privilegios para el uso de espacio en disco y la conectividad a todas las impresoras. Revisar drivers y archivos

necesarios para nuestras impresoras pueden comunicarse con el sistema de impresión instalado.

### **Recomendaciones**

Durante la migración será útil revisar los siguientes aspectos:

- Revisar compatibilidad desde todas las plataformas existentes
- Revisar compatibilidad desde todas el software en uso
- Revisar permisos de impresión
- Revisar cantidad de hojas autorizadas para la impresión
- Espacio en disco del servidor para que el flujo de trabajos de impresión pueda ser almacenado

### **3.2.5 Respaldo**

#### **Tecnologías**

- **Amanda**

Advanced Maryland Automatic Network Disk Archiver, es un sistema de respaldo automatizado que permite el respaldo de múltiples computadores sobre la red. El destino puede ser un disco o cintas o dispositivos ópticos. Amanda utiliza herramientas GNU y puede realizar respaldos en Linux y MS-Windows.

#### **Migración**

En Linux bastará con instalar la aplicación y configurar el software de acuerdo a las necesidades. La migración sobre MS-Windows requerirá la instalación adicional de cygwin.

Para respaldos realizados con otras herramientas, una alternativa podrá ser la definición de un directorio de paso para recuperar y luego respaldar con Amanda.

### **Recomendaciones**

Cuando adoptamos un nuevo sistema de respaldo, es probable que no conservemos todos los atributos. Esto dependerá de las variables que logre capturar desde el origen de los datos. Si el software de respaldo original no capturó permisos, fechas u otros datos relevantes, este trabajo deberá realizarse en forma manual. El principal riesgo será que usuarios no autorizados accedan a los respaldos recuperados.

### 3.3 Mensajería

#### 3.3.1 Servicio de Correo

El correo es un área compleja con varios componentes , y existen diversas aplicaciones FLOSS, algunas de las cuales proporcionan una funcionalidad parcialmente coincidente.

Sin embargo no hay que olvidar que el tratamiento de correo no esta directamente relacionado en el manejo de SPAM y virus para computadores. Por lo tanto es necesario considerar el manejo de correo independientemente del tratamiento de amenazas para la productividad transportadas en este mismo.

#### **Tecnologías**

Los sistemas de correo están subdivididos de acuerdo a sus funcionalidades. En esta sección revisaremos los sistemas de transporte y almacenamiento de correo.

- **MTA**

Los principales MTA de FLOSS son:

- Sendmail
- Exim
- Courier
- Postfix

Hay muchos más pero estos son los más utilizados. En un principio Sendmail fue el más utilizado. Sin embargo su escaso nivel de seguridad y elevado nivel de conocimientos para configurarlo, lo transformó en un software mítico y poco utilizable. Posteriormente aparecieron en sitios de divulgación y listas administradores otros agentes. En estos agentes se muestra apoyo diferente y de mejor sofisticación para cada nicho de usuarios. Exim y Postfix son el MTA por defecto en algunas distribuciones. Courier-MTA viene como parte de una familia formada por MTA, MDA (Mail Delivery Agent), MUA y un paquete de correo en web. Cada una de estas partes se puede usar por separado, o integrada con el resto de la familia. Uno de los aspectos más relevantes será el sistema de almacenamiento que usualmente son archivos Maildir o en archivos Mailbox para cada usuario.

- **Buzón**

La mayoría de los administradores desearían que los clientes usaran un buzón de correo centralizado en lugar de que descarguen los mensajes a un buzón local en el cliente de escritorio. Por ello recomendamos el uso de IMAP como protocolo de intercambio.



Servidores IMAP FLOSS:

- UW-IMAP
- Courier-IMAP
- Cyrus.

UW-IMAP ha mostrado diversas vulnerabilidades por lo que hay que evaluar bien antes de tomar la decisión de migración.

Courier-IMAP destaca por su facilidad de configuración. Es posible instalarlo junto a Postfix o Courier-MTA.

Courier como plataforma implementa estándares que en ocasiones no se utilizan en el transporte de correo. Por este motivo es recomendable revisar los foros.

Tanto Courier-IMAP como Cyrus soportan TLS .

Adicionalmente al manejo de correo electrónico se ha incorporado la protección contra SPAM (Correo no deseado).

Si reemplazamos toda la plataforma de correo sería deseable introducir un antispam como Spamassassin.

SpamAssassin es un filtro automático de correo que revisa los encabezados (header) y el cuerpo de los mensajes (body) para clasificar los correos electrónicos que posiblemente sean SPAM en base a una reglas bastantes automáticas y configurables por el administrador.

El resultado se traduce en una disminución de correo no deseado, recuperando tiempo de trabajo al usuario final.

## **Migración**

Distinguir el sistema de almacenamiento es fundamental para poder ejecutar una migración adecuada. El sistema de almacenamiento es la forma en que la aplicación guarda los correos en un sistema de archivos o base de datos. Generalmente nos encontraremos con un Mailbox o archivo, que puede ser plano, o con Maildir. Maildir es un sistema de directorio que posee tres carpetas para ordenar el correo de un usuario en particular.

Aplicaciones para convertir de un sistema a otro se pueden encontrar. La conversión más común es desde Mailbox a Maildir. El sistema de correo Sendmail utiliza Mailbox. Si queremos mover los archivos a nuestros nuevos Maildir, encontraremos diversos *scripts* que nos ayudarán en esta tarea.

## **Recomendaciones**

Debido a que el servicio de correo funciona en forma continua, es recomendable tomar las siguientes precauciones.

- Verificar que no se interrumpa la recepción de correo
- Habilitar una pasarela para garantizar el paso de correo y almacenamiento en forma transitoria
- Almacenar el registro de logs para evitar pérdidas de correo
- Verificar el espacio para almacenar el correo de la organización
- Establecer procedimientos e implementar herramientas para el control del espacio almacenado por cada usuario

### **3.3.2 Servicios de Correo WEB**

#### **Squirrelmail**

Disponible para plataformas que posean PHP 4.0 o superior y un servidor de acceso IMAP.

#### **Horde IMP**

Plataforma desarrollada en PHP que es parte del framework Horde. El objetivo principal de Horde es proveer una plataforma basa en estándares sólidos. Por este motivo será integrable mediante protocolos como POP3 e IMAP a los servidores de correo.

#### **Migración**

La migración consistirá en la instalación de este paquete. Para aquellos administradores que migren de plataformas propietarias, esta interfaz será un poco menos usable y con menos funcionalidades que una agenda.

#### **Recomendaciones.**

Deseable sería la personalización de la interfaz de correo. Esto les dará a los usuarios una mejor experiencia de uso. En términos prácticos se reducirá la carga de la mesa de ayuda y los usuarios podrán en forma más amigable el nuevo sistema.

Debemos estar preparados para recibir críticas que compararán servicios webmail públicos con nuestro nuevo sistema de correo.

## **3.4 Servicios seguridad**

### **3.4.1 Identidad electrónica**

En un problema de difícil solución, el administrador se ve enfrentado a poder asegurar con los medios existentes la identidad de un usuario. Si bien, las directrices para la implementación electrónica de la identidad electrónica están bien definidas, usualmente los

problemas no son causa de una migración, sino de la captura de los datos biométricos de los usuarios.

### **3.4.2 Autenticación centralizada**

Componente indispensable para el desarrollo de procesos en donde intervienen diversos componentes. Durante una migración, el aspecto vital es rescatar usuarios y perfiles aplicativos para cada aplicativo. Una migración de perfiles es distinguible en dos partes. La primera involucra migrar de un sistema propietario hacia un sistema abierto. La segunda es involucrar la tarea de aglomerar las políticas de acceso en un mismo sistema. En este sentido, muchas veces se confunde la tarea de migración y la tarea de unificación. Esta guía solo contempla el aspecto de migración y por este motivo las tareas asociadas a la integración deberán ser tratadas como otro aspecto.

Para poder distinguir las tareas de migración e integración solamente es necesario revisar el conjunto de salida y llegada de un dominio de usuarios en particular. Si el dominio de llegada es mas amplio que el sistema único de migración, estaremos enfrentados a la tarea de integración o adición de políticas de usuarios.

El estándar más popular para servicios de directorios es el LDAP. Consiste en un protocolo abierto y se incluye en muchos productos y en diferentes ámbitos LDAP trabaja con definiciones de datos personalizados. Los datos almacenados en esquemas son generalmente distintos. Por lo cual no es suficiente para la migración de una aplicación, el estado de cliente de LDAP. Es necesario comparar los esquemas y eventualmente realizar un trabajo basado en scripts para transformar los datos almacenados en LDAP para que sean compatibles con nuestra nueva aplicación.

#### **OpenLDAP**

Utiliza el estándar LDAPv3, y a partir de la versión 2.1 se puede utilizar con diversos sistemas para la generación. Las mismas consideraciones mencionadas anteriormente son válidas para OpenLDAP. Lo principal es el esquema de utilización para una aplicación en particular. En una instalación, cuya instalación se hace desde el diseño del almacenamiento de usuarios, será viable y mucho más fácil.

### **3.4.3 Antivirus**

La migración de un antivirus es meramente funcional. Esto implica que solo es necesario activar el nuevo sistema para iniciar la detección de amenazas. El problema actual es poder medir si la cantidad de amenazas cubiertas es suficiente para poder seguir operando de manera normal y confiable.

## Tecnología

- **ClamAV**

Dispone de un número de versiones para sistemas operativos como Linux, BSD, OpenSolaris, MS-Windows entre otros. La detección de virus no es proactiva. Lo que implica que debe ser bajo demanda. A diferencia de la mayoría de los paquetes de antivirus comerciales.

La clasificación de ClamAV es la siguiente:

- Bajo demanda : muy pobre
- Falsos positivos: pobre
- Proactiva: pobre
- Tiempo de respuesta: Muy bueno
- Rootkits: muy pobre

- **Amavis**

Motor para detección de virus sobre plataformas de correo FLOSS. Amavis soporta firmas de otros fabricantes de antivirus.

Funciona sobre las siguientes plataformas de correo.

- Sendmail
- QMail
- Postfix
- Exim

## Migración

En el caso de un PC de escritorio consistirá simplemente en desinstalar el paquete de antivirus existente, en caso que exista alguno e instalar el paquete.

## Recomendaciones

El mejor uso para el antivirus FLOSS es una pasarela (Gateway) de correo. El reemplazo de una de estos paquetes podría crear la falsa sensación de seguridad, debido a que el análisis automático en una estación de trabajo deberá hacerse bajo demanda.

### 3.4.4 VPN

## Tecnologías

- **FreeSWAN**

Es una aplicación GNU/Linux de los estándares IPSEC e IKE, lo que significa que

operarán recíprocamente con los dispositivos compatibles concentradores de VPN y otros elementos de red. En el caso que el sistema de VPN se haya configurado para IPv6 los registros del DNS deberán ser actualizados, lo que podría implicar ciertas limitaciones.

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

## **Tecnologías**

### **Migración**

Verificar la Interoperabilidad con las plataformas interconectadas será la tarea más costosa. Deberá ser parte de la etapa de pruebas para poder tomar la decisión de implantar cualquier sistema de VPN, Clientes propietarios podrán ser reemplazados si existen las condiciones de interoperabilidad mencionadas anteriormente.

### **Recomendaciones**

Pruebas desde todos los clientes serán necesarias para soportar el cambio. Además pruebas de carga con usuarios concurrentes, para verificar que el hardware soporte los niveles de cifrado o encriptación requeridos.

#### **3.4.5 Firewall**

### **Tecnologías**

Actualmente podemos encontrar Packet Filter de BSD e Iptables disponible para la plataforma Linux. Dependerá de la versión del sistema operativo que instalemos. Todas las interfaces de acceso a la configuración del firewall son herramientas complementarias. La existencia de estas aplicaciones en entornos gráficos o textuales se debe a la complejidad para generar configuraciones en base políticas.

Existen distribuciones dedicadas a proveer un Firewall. Estas distribuciones se encuentran disponibles en su mayoría para dedicar un computador para esta tarea.

### **Migración**

La traducción de reglas será fundamental. Fuera de trabajar con un especialista, la mejor

alternativa para la migración será conocer las políticas e implementarlas. Una práctica poco deseable será traducir la configuración debido a que puede inducir a error, olvidando la política original y además puede basarse en alguna particularidad del firewall anterior.

### **Recomendaciones**

Dedicar un equipo completamente a esta tarea. Normalmente deberemos elegir un equipo que soporte todo el tráfico de nuestra red hacia Internet. Por este motivo probar con diferentes protocolos, casos de uso y casos concurrentes será indispensable. Además utilizar herramientas para realizar revisión de puertos abiertos será de utilidad para revisar que se cumplan las políticas implementadas.

## **4 Consideraciones**

Todo el trabajo de migración se basará principalmente en la definición de la metodología a emplear. Las condiciones de software y hardware cambiarán muchas veces en el futuro. Sin embargo la organización de los recursos humanos y la gestión deberán ser siempre parte de un proceso de migración. Es muy probable que la complejidad del software aumente con el tiempo. Quizás también encontremos muchas herramientas que nos facilitarán el proceso. Por este motivo todas las consideraciones de software cambiarán con el correr del tiempo. Pero la realización de un plan de trabajo acotado y bien documentado nos facilitará la tarea de migración. Debemos tener en cuenta que la utilización de estándares abiertos también nos ayudará a integrar de mejor manera herramientas FLOSS a nuestro ambiente de trabajo.

## 5 Glosario

El presente glosario puede ser útil para entender posibles siglas que involucren software o tecnologías necesarias en el proceso de migración.

**AFS** Archivos de Sistema Andrew (Andrew File System)

**ASP** Active Server Pages - Páginas de Servidor Activas; es el concepto de Microsoft™ para generar sitios web dinámicos (consultar también “JSP”) del lado del servidor (usando, por ejemplo, JavaScript o Visual Basic Script).

**BSD** Distribución de Software de Berkeley (Berkeley Software Distribution)

**CGI** La interfaz Común de Entrada (Common Gateway Interface) fue la primera variante de las interfaces de servidores web. Prácticamente todos los servidores web modernos soportan esta interfaz. Las aplicaciones que usan CGI pueden ser desarrolladas en diferentes lenguajes de programación. Además de lenguajes interpretados, como PERL, también es posible utilizar aplicaciones compiladas que hayan sido escritas en C o C++.

**CUPS** Sistema de Impresión Común UNIX (Common UNIX Printing System)

**DAV** Distributed Authoring and Versioning

**DHCP** El Protocolo de Configuración Dinámica de Anfitrión (Dynamic Host Configuration Protocol) crea las bases para la asignación dinámica de direcciones IP. El cliente DHCP dinámicamente recibe una dirección IP desde un servidor DHCP central. Además de la dirección IP, otros parámetros de configuración pueden ser enviados al cliente.

**DNS** El Sistema de Nombres de Dominio (Domain Name Server) es un sistema con una estructura jerárquica para la asignación de nombres a las computadoras conectadas a Internet/Intranet.

**GNU** GNU no es Unix (GNU's Not UNIX)

**GPL** Licencia Publica General (General/Gnu Public License)

**HTML** Lenguaje de Marcas de Hipertexto (Hypertext Markup Language) - Es el estándar abierto y el formato de archivo para la presentación de contenidos en Internet e Intranets.

**HTTP** Hypertext transfer protocol Un estándar para interacción electrónica durante la transmisión de documentos web en Internet.

**IMAP** El Protocolo de Acceso a Correo de Internet (Internet Mail Access Protocol) puede ser usado para administrar casilla de e-mail. A diferencia de POP3, IMAP administra los e-mails en el servidor. Cuando el programa de correo comienza, solo los datos de encabezado (remitente, referencia y fecha de recepción) se carga por defecto. El receptor puede entonces seleccionar los mails a bajar en forma completa. Los correos que quedan en el servidor

pueden ser archivados en carpetas especiales.

**IPSec** Un estándar para soluciones de seguridad en redes que es particularmente adecuada para la implementación de VPNs y para acceso remoto a redes privadas por medio de conexiones telefónicas.

**IPv6** La nueva versión 6 del Protocolo de Internet (IP) con direcciones IP de 128 bits en lugar de los 32 bits de Ipv4. Esto puede crear más opciones de direccionamiento para sitios web.

**Java** Un lenguaje de programación desarrollado por SUN Microsystems™ que es usado especialmente en el campo de la tecnología de Internet. Un así llamado compilador traduce el texto fuente a código independiente de la plataforma. El código entonces puede ser ejecutado por un interprete adecuado en cualquier computadora. Esto permite la ejecución de programas Java en cualquier plataforma de computación para la que exista el programa interprete adecuado.

**JavaScript** Un lenguaje de scripting originalmente definido por Netscape para conectar código de programa a páginas HTML estáticas. El código es generalmente ejecutado en el browser del usuario.

**J2EE** Edición empresarial de Java2 (Java 2 Enterprise Edition)

**JDBC** La Conectividad a Base de Datos Java (Java Database Connectivity) ofrece un mecanismo para comunicarse con las bases de datos existentes. Los drivers sirven como la interfaz entre Java y la base de datos.

**JSP** Las Páginas de Servidor Java (Java Server Pages) son archivos HTML con código de programa Java embebido que se convierte en servlets por un motor JSP que son luego ejecutados en el servidor web. El resultado se envía luego en formato HTML normal al cliente (consultar también "ASP").

**LAMP** Un plataforma de software libre para desarrolladores web y aplicaciones web basado en Linux, Apache, MySQL y PHP y/o PERL o Python.

**LDAP** El Protocolo Liviano de Acceso a Directorio(Lightweight Directory Access Protocol) (X.509) es una versión simplificada de DAP (X.500). LDAP se usa para acceder a servicios de directorio que pueden, por ejemplo, ser usados para consultar características de los usuarios.

**LPD** Demonio de impresión en línea (Line Printing Daemon)

**LPR** Redirector de impresiones en línea (Line Printing Redirector)

**Macro** Una combinación de instrucciones individuales y/o una secuencia de comandos y procesos que pueden ser grabados y guardados. Cuando se llama una macro, los procesos y acciones se ejecutan automáticamente en el orden correcto.

**MTA** Un componente de software responsable de la distribución de e-mails entre diferentes sistemas de computación. Un MTA recibe mensajes de otros MTAs y de MUAs y los pasa a



los correspondientes recipientes.

**MUA** El Agente de Usuario de Correo (Mail User Agent) es el programa de e-mail que permite a los usuarios acceder, mostrar, leer, editar y administrar mensajes electrónicos.

**.NET** Una plataforma para servicios web basados en XML de Microsoft™. La plataforma se diseñó para conectar información, dispositivos y usuarios de forma uniforme y personalizada.

**NAT** Network Address Translation

**NFS** Network File System

**NTP** El Protocolo de Tiempo en Red (Network Time Protocol) se emplea para sincronizar la información horaria de diferentes ordenadores usando la red. NTP permite sincronizar los ordenadores al orden de milisegundos. Esto es un aspecto particularmente importante para procesos en los que están involucrados varios ordenadores a la vez.

**ODBC** Conectividad a Bases de Datos Abierta (Open DataBase Access). Es un proceso de estandarización del acceso a bases de datos. Las aplicaciones, por ejemplo, pueden usar ODBC para acceder a un gran número de bases de datos de diferentes fabricantes.

**OLE** Enlazado e incrustación de objetos (Object Linking and Embedding). Es un método para compartir información. Esta información puede estar representada en diferentes formatos y puede haber sido generada por diferentes aplicaciones. Los datos de un documento fuente se enlazan o incrustan en un documento destino. Cuando se marcan los contenidos en el documento destino, la aplicación origen se ejecuta, por lo que los datos pueden editarse utilizando las herramientas disponibles en el entorno original.

**PDF** Formato de Documento Portable (Portable Document Format). Un formato de documento legible en varias plataformas, desarrollado por Adobe Systems™, que permite la generación y presentación de documentos formados por texto, imágenes y dibujos.

**Perl** Lenguaje Práctico de Extracción e Informe (Practical Extraction and Report Language). El lenguaje Perl está disponible de manera libre y está ampliamente utilizado en la escritura de scripts CGI. Gracias a la cantidad de opciones, especialmente en el tratamiento y procesamiento de cadenas de texto, los programas Perl también se emplean en tareas rutinarias de administración.

**PHP** Preprocesador de Hipertexto (Hypertext Preprocessor). Lenguaje de script del lado servidor para la generación de contenido dinámico y acceso a bases de datos.

**POP/POP3** Protocolo de Oficina de Correos (Post Office Protocol). Cuando se emplea la versión 3 de este protocolo de correo electrónico, el cliente de correo descarga todos los nuevos mensajes desde el servidor de correo a la máquina local. Los clientes suelen estar configurados para eliminar los mensajes del servidor una vez se han descargado correctamente.

**PostScript** lenguaje de descripción de páginas desarrollado por Adobe™ para controlar las impresoras. Las impresoras que admiten PostScript reciben los comandos de impresión desde los programas en forma de una secuencia estandarizada de instrucciones, que la

impresora interpreta y traduce a un proceso de impresión. RAS: Servicio de acceso remoto (Remote Access Service). Microsoft™ utiliza este nombre para proporcionar servicios basados en llamadas de teléfono dentro del sistema operativo de Microsoft™.

RDBMS Sistema de Gestión de Bases de Datos Relacionales (Relational DataBase Management System). Los datos de una base de datos en un RDBMS se almacena en tablas, que están relacionadas unas con otras. Esta organización se basa en un modelo relacional.

S/MIME Secure MIME (Multipurpose Internet Mail Extensions)

SMB Server Message Block

SQL Lenguaje de Consulta Estructurado (Structured Querying Language). El lenguaje de interrogación estándar para las base de datos relacionales.

SSL/TLS Secure Socket Layer. Una tecnología de cifrado desarrollada por Netscape™ y un protocolo para la comunicación segura y la transmisión de documentos entre navegadores web y servidores web.

TLS Transport Layer Security (ver SSL/TLS).

VPN Virtual Private Network

W3C World Wide Web Consortium. Consorcio que coordina el desarrollo de la web y la estandarización de HTML, XML y sus derivados.

WebDAV Control de versiones y Autoría Distribuida basada en la web (Web-based Distributed Authoring and Versioning). Este protocolo es una extensión del Protocolo para la Transferencia de Hipertexto (HTTP - HyperText Transfer Protocol) y ofrece un soporte estandarizado para la una creación de contenidos asíncrona y en colaboración a través de Internet o Intranet.

WWW World Wide Web

XML eXtensible Markup Language. Una especificación en la definición de lenguajes para formatear documentos. XML ofrece una estricta separación de contenidos y diseño.