



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

---

**FACULTAD DE ESTUDIOS SUPERIORES**

**CAMPUS ARAGON**

**METODOLOGIA DE MIGRACION EN TIEMPO REAL,  
CASO DGSCA**

**T E S I S**

**PARA OBTENER EL TÍTULO DE:  
INGENIERO EN COMPUTACION**

**PRESENTA:  
RAMÓN GALEANA HUERTA**

**ASESOR:  
ING RODOLFO VAZQUEZ MORALES**

**MÉXICO, D. F.**

**2010**





Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*A mi madre, quien me enseñó que todo sacrificio es bueno*

*A mi hermana, por su apoyo incondicional*

*Para la familia que me tocó, para la familia que escogí*

# ÍNDICE

Introducción . . . . .	9
Capítulo 1. Esquemas de migración . . . . .	13
1.1 Que es una migración informática . . . . .	13
1.2 Problemática de migración de aplicaciones . . . . .	13
1.3 Esquemas de migración . . . . .	15
1.4 Etapas de un sistema de migración . . . . .	16
1.4.1 Análisis de Impactos en la organización . . . . .	17
1.4.2 Descubrimiento . . . . .	17
1.4.3 Diseño y mapeo . . . . .	18
1.4.4 Creación de un plan de migración . . . . .	18
1.4.5 Previsiones . . . . .	19
1.4.6 Pruebas antes de la migración . . . . .	20
1.4.7 Migración y cutover . . . . .	20
1.4.8 Validación de la migración . . . . .	21
1.5 Metodología en tiempo real . . . . .	23
1.5.1 Etapa de descubrimiento . . . . .	23
1.5.1.1 Identificar todos los elementos técnicos involucrados . . . . .	24
1.5.1.2 Desglosar cada elemento y su configuración actual . . . . .	24
1.5.1.3 Ubicar a cada elemento dentro del proceso general de datos . . . . .	24
1.5.1.4 Identificar cuales elementos son los que tienen menor y mayor concurrencia y menor tasa de cambios . . . . .	25
1.5.1.5 Identificar las configuraciones internas y externas . . . . .	25
1.5.2 Diseño y mapeo . . . . .	25
1.5.2.1 Identificar posibles sustitutos para cada elemento en caso de ser necesario . . . . .	25

1.5.2.2 Ubicar aquellos elementos que pueda redirigir el flujo de datos. . . . .	26
1.5.3 Pruebas . . . . .	26
1.5.3.1 Instalar en el equipo nuevo cada una de las partes . . . . .	27
1.5.3.2 Realizar pruebas del funcionamiento de cada una de las partes. . . . .	27
1.5.4 Migración y cutover . . . . .	27
1.5.4.1 Migrar elementos en orden de actividad . . . . .	27
1.5.4.2 En la medida de lo posible utilizar o diseñar y programar una herramienta que automatice el proceso . . . . .	28
1.5.4.3 Realizar un simulacro de migración . . . . .	28
1.5.5 Validación de la migración . . . . .	28
1.5.5.1 Verificar la totalidad del proceso . . . . .	28
1.5.5.2 Realizar los últimos cambios si así se requiere . . . . .	28
Capítulo 2. Elementos de operación para la migración . . . . .	31
2.1 Problemática . . . . .	31
2.2 Antecedentes . . . . .	32
2.3 Migración de Usuarios . . . . .	34
2.4 Migración de Correos. . . . .	34
2.5 Operación . . . . .	35
2.5.1 LDAP . . . . .	35
2.5.2 DaemonTools . . . . .	37
2.5.2.1 Características. . . . .	38
2.5.2.2 Cómo funciona supervise . . . . .	38
2.5.2.3 Script Run y UCSPI (Unix Client-Server Program Interface) . . . . .	39
2.5.3 Qmail y SMTP (Simple Mail Transfer Protocol). . . . .	40
2.5.4 IMAP Y POP. . . . .	40
2.5.5 Webmail Horde. . . . .	41
2.6. Funcionamiento de Neon antes de la Migración . . . . .	41
2.6.1 Correo de entrada . . . . .	41
2.6.2 Correo de salida . . . . .	42
2.6.3 Operación de Webmail Horde. . . . .	42
Capítulo 3. Proceso de migración. . . . .	45
3.1 Proceso de migración. . . . .	45

3.2 Detalles de Xalumno. . . . .	45
3.3 Consulta de Correo. . . . .	48
3.4 Transferencia de Correo. . . . .	48
Capítulo 4. Configuración de pruebas . . . . .	51
Instalación . . . . .	51
4.1 Instalación Qmail-Ldap. . . . .	51
4.1.1 Ventajas. . . . .	51
4.1.2 Desventajas. . . . .	51
4.2 Instalación Linux Red Hat AS 3. . . . .	52
4.3 LDAP. . . . .	52
4.3.1 Configuración . . . . .	52
4.3.2 Comandos ldap de administración . . . . .	55
4.4 Qmail . . . . .	59
4.4.1 Instalar Qmail . . . . .	60
4.4.2 Configuración de Qmail . . . . .	62
4.4.3 Configurar qmail SMTP service . . . . .	66
4.5 Instalación de Courier-IMAP . . . . .	67
4.6 Configurar servicios SSL . . . . .	72
4.6.1 POP3SSL . . . . .	73
4.6.2 IMAPSSL . . . . .	74
4.7 Archivos de configuración del sistema de correo . . . . .	75
Capítulo 5. Implementación . . . . .	81
Migración. . . . .	81
5.1 Migración de LDAP . . . . .	81
5.2 Envío de correo. . . . .	85
5.3 Consulta de correo . . . . .	86
5.4 Migración de correo antiguos mediante llaves SSH. . . . .	87
En Neon . . . . .	88
5.5 Script de Migración. . . . .	89
Conclusiones . . . . .	93
Bibliografía . . . . .	95

Anexo .....	97
Estructura de un correo electrónico .....	97
Introducción al Protocolo IMAP.....	99
Introducción al protocolo POP.....	101
Introducción al protocolo SMTP .....	103
Intercambio de correo .....	103
Flujo de transacción de correo de SMTP.....	104

## INTRODUCCIÓN

**E**l uso y diversidad de infraestructura de tecnología informática ha tenido un gran crecimiento en todas las instituciones de nuestro país y del mundo, el desarrollo de nuevas tecnologías y nuevos protocolos han impulsado el incremento de aplicaciones y servicios nuevos e innovadores, sin embargo, este avance ha tenido sus complicaciones, ya que se han tomado en cuenta aspectos como: crecimiento, compatibilidad, asimilación tecnológica, y operación.

El crecimiento de nuestros usuarios, información y aplicaciones, nos obliga muchas veces a migrar dichas aplicaciones e información a equipos de mayores capacidades para obtener un mejor desempeño y productividad en los usuarios que utilizan los sistemas de un empresa o institución, pero este tipo de actividades trae una serie de inconvenientes como son la continuidad de servicio y el formateo de datos dentro del nuevo esquema, lo que provoca que se planteen esquemas secundarios de operación mientras se realiza el cambio de un nuevo equipo.

Muchas son las aplicaciones que han rebasado las capacidades de los equipos donde se ejecutan, no podemos decir que es un error de requerimientos iniciales, pero sí que no se contemplo objetivamente el crecimiento de los usuarios, información, tecnologías y sus correspondientes configuraciones.

Para los administradores de tecnologías de la Información las migraciones de datos y sistemas se han convertido en una actividad usual y muchas veces con enormes retos dentro de su campo laboral. La carga de trabajo, la actualización de software y tecnología, la relocalización de centros de almacenamiento de datos, la implementación de esquemas de seguridad conducen a mover sistemas y datos de un dispositivo a otro teniendo como trasfondo la configuración de las nuevas o actualizadas herramientas dentro de los nuevos esquemas



e incluso la asimilación de nuevos conceptos y tecnologías que se pueden aplicar al nuevo sistema.

Las migraciones pueden llegar a ser una actividad rutinaria y monótona si se realizan con frecuencia en entornos ya conocidos, pero esto no implica en lo absoluto que sea un procedimiento sencillo de repetir una y otra vez en otros escenarios ya que estos cambian en función de las necesidades y características del ambiente al que se quiere migrar. Las migraciones (mover datos y sistemas de un dispositivo a otro) y la redirección de todas las operaciones de entrada y salida al nuevo dispositivo es un proceso laborioso ya que es necesario conocer a fondo la totalidad del sistema y por ende su flujo de datos.

Con el incremento en el porcentaje de procesamiento de datos de misión crítica y las demandas de alta disponibilidad de estos datos, el proceso de migración muchas veces encara la necesidad de realizar una baja momentánea (lapsos que van desde minutos a horas e incluso uno o dos días de inactividad) de sistemas lo que trae consigo un gran impacto en la institución que repercuten negativamente en su imagen pública ya que vulnera la propiedad de confianza que la empresa pueda tener y que repercute también en el aspecto económico.

Aunado a esto el proceso de migración se ve acotado también por los requerimientos del negocio, requerimientos técnicos, y operacionales e incluso requerimientos humanos subjetivos que hacen del proceso una tarea muchas veces prolongada e incluso tediosa. La demanda de recursos computacionales, los riesgos asociados a ellos tales como la corrupción o pérdida de datos, incompatibilidades técnicas, el tiempo en que estará fuera de funcionamiento el sistema, y el impacto en el desempeño de este hacen que el proceso de migración se convierta en un reto muchas veces difíciles de abordar y plantear para llevarla a cabo satisfactoriamente.

Dentro del mundo del manejo de tecnologías de información existe una amplia variedad de metodologías que abordan este tipo de situaciones que dependiendo de la importancia de la información que maneja un sistema, su concurrencia, su desempeño y su disponibilidad se pueden ser correctamente aplicables pero no sin antes conocer las potenciales desventajas que trae el aplicarlas.

Esta decisión debe estar correctamente respaldada por un estudio previo que analice los pros y los contras de cada caso en específico para disminuir al máximo el riesgo de pérdida de información, de inactividad de servicios y la correspon-

diente indisponibilidad de estos valores hacia las personas o sistemas que hacen uso de ellos dañando con la reputación que la institución pudiera tener.

El objetivo de la presente tesis es la propuesta de una metodología para los procesos de migración de sistemas en tiempo real, la cual se demostrará implementándola en la aplicación del servicio de correo de exalumno dentro de la UNAM.

Dentro del primer capítulo se explicaran el concepto de migración, las principales causas que originan un proceso de migración, los esquemas de migración mas comúnmente usados ,sus características y sus principales desventajas, las etapas que debe contar un proceso de migración. Se desarrollaran los puntos que involucran una migración en tiempo real y las variables a considerar en cada punto crítico.

Dentro del segundo capítulo se explica la problemática y necesidades de la migración del servidor de correo electrónico dentro del departamento de Administración de Servidores de la dirección de Telecomunicaciones en la dirección General de Computo Académico dependencia ubicada en la Universidad Nacional Autónoma de México, los antecedentes de migración dentro de este departamento así como una descripción de los componentes que integran este servicio lo que corresponde al primer punto de la metodología.

En el capítulo tres se describe el funcionamiento del flujo de información entre los componentes del servicio de correo y los pasos a nivel general que se tuvieron que tomar en cuenta para el proceso de migración cubriendo así los puntos de descubrimiento, diseño y mapeo de la metodología general de migración.

Los puntos referentes a la creación de un plan de migración, previsiones y pruebas antes de la migración se detallan en el capítulo cuatro, es decir se describe la instalación de cada uno de los componentes, servicios y herramientas involucrados en el nuevo equipo y sus correspondientes configuraciones y pruebas para que su funcionamiento sea igual al equipo a punto de migrar.

Una vez que se tienen las herramientas ya instaladas y probadas se procede finalmente a la migración de cada uno de los elementos que procesaran la recepción, consulta y envío de correo así como los contratiempos de cada uno de ellos es decir la validación de la migración, todo esto explicado en el quinto y último capítulo

# CAPÍTULO 1

## ESQUEMAS DE MIGRACIÓN

### 1.1 Que es una migración informática

Una migración informática consiste en la transferencia de datos o programas que funcionan en una computadora o servidor a otro de mejores características con el fin de incrementar o reforzar algunos puntos críticos dentro de su funcionamiento o para resolver uno o varios problemas que se presentan a continuación.

### 1.2 Problemática de migración de aplicaciones

La pregunta fundamental que un administrador de alguna aplicación corriendo en un ambiente empresarial se hace cuando se le encomienda una tarea de migración de datos o aplicación es el porque de esa migración. La respuesta se puede resumir en los siguientes puntos;

- Mantenimiento.
- Obsolescencia.
- Lentitud.
- Costo.
- Crecimiento.

El mantenimiento de un sistema regularmente se enfoca en la actualización e instalación de parches a distintas partes de una o varias aplicaciones, la agre-

gación de varios módulos que incrementan las funcionalidades de la misma y que no formaban parte de su objetivo inicial y en el cual no hubo un rediseño natural que permitiera aprovechar las ventajas de algún paradigma de programación y acabaran por enclavarse de manera forzada lo que la gran mayoría de las veces se ve reflejado en su desempeño , su documentación, y en la dificultad de realizar alguna corrección, o modificación ad-hoc, o legibilidad de su código.

La obsolescencia se refiere a la inutilidad actual de la aplicación, ya que los objetivos con los que fue creado han sido expandidos superando incluso las capacidades de este o simplemente porque cambiaron de rumbo, las características del lenguaje de programación con el que hizo no soportan la nuevas exigencias de quienes lo utilizan, el sistema operativo requiere de nuevos componentes para los que ya no hay versiones ni soporte, causas que exigen una actualización del sistema e incluso la creación de uno mas adecuado a las nuevas necesidades para el que es requerido.

La lentitud se genera por la degradación natural del equipo de computo en uso ya sea en su procesador, discos duros y memoria en el que se aloja la aplicación, la incapacidad de los parches para corregir puntos críticos, el despliegue de otras aplicaciones que no estaban contempladas en la instalación del sistema, la expansión de otros equipos dentro de la red y al crecimiento de usuarios que utilizan dicha aplicación son hechos que afectan su rendimiento.

Los costos de mantenimiento, actualización y de lentitud pueden hacer que el hospedaje de la aplicación ya no sea viables o redituables para la institución o negocio, o que existan alternativas que son mucho mas baratas e incluso gratis que, que aunque requieren de soporte y personal capacitado para su operación y administración no se compara con el gasto de una licencia propietaria.

Este tipo de gastos hacen que el sistema o aplicación en cuestión sea un posible candidato para su desaparición o renovación.

Como se menciono anteriormente el crecimiento de usuarios que accedan a una aplicación conlleva a la necesidad de adquirir un equipo de mayores dimensiones e incluso a implementar algún esquema de aplicaciones distribuidas o clustering para el manejo de carga de trabajo o espejeo de aplicaciones para una mayor disponibilidad del sistema.

### 1.3 Esquemas de migración

Los distintos esquemas de migración que existen son los siguientes;

- Pasiva
- Por Partes
- En tiempo real

La migración pasiva es aquella donde existen dos o más computadoras, una con el sistema viejo y el otro con el sistema más reciente, con los mismos elementos y en donde la computadora nueva tiene ya instalado las nuevas versiones de la aplicación y se han realizado pruebas y depuración de errores sin que estas actividades modifiquen el funcionamiento del antiguo sistema como sucede en otros esquemas y por lo tanto es menos propensa a errores, la desventaja de este esquema radica en que para realizar con éxito la transición del viejo sistema al nuevo es necesario dar de baja temporal o sacar de línea a su predecesor lo que afecta la imagen del negocio y/o deja ociosos a los usuarios en este periodo de tiempo que no siempre es el que se ha establecido.

En la migración por partes se migran primeramente aquellos elementos cuya modificación de datos no es tan frecuente y donde se le puede dar de baja en un marco de tiempo muy corto o inexistente, y de esta manera hacer que los elementos residentes en el viejo sistema apunten a los nuevos para poder dar de baja a los viejos elementos, y enseguida migrar en orden de adhesión a los otros elementos con lo que el intervalo de tiempo de inactividad se reduce para aquellos elementos que son críticos en la operación y que cambian su información de manera constante.

En la migración en tiempo real la transición de todo el sistema es transparente para el usuario ya que la computadora en donde se alojara el nuevo sistema cuenta con los elementos ya probados y configurados adecuadamente, el problema de este tipo de migración radica en que hay que realizar un análisis minucioso de cada una de sus partes y el cómo interactúa cada una de estas partes con las demás, es decir conocer el flujo de datos que ocurre cuando existe una petición o envío de información en cada una de sus partes, esto se

hace con la finalidad de poder redirigir ese flujo a los componentes correspondientes en el nuevo sistema y así evitar la baja temporal de ese componente, otro problema radica en los planes de contingencia que se pueden presentar cuando existe un error en el proceso de migración y que nos permitan recuperarnos rápidamente de estos errores o incongruencias. La principal ventaja de este tipo de migración es la disponibilidad de datos y servicios se reduce drásticamente e incluso se elimina en comparación con otros los anteriores tipos de migración.

## 1.4 Etapas de un sistema de migración

Aunque el objetivo final de una migración es trasladar información y programas es necesario que exista una planificación previa para que dicha migración se realice con éxito independientemente de su complejidad. La planificación no solo es una ayuda durante el proceso de migración sino que también reduce los impactos y riesgos negativos que pueden afectar al negocio o institución (por ejemplo el tiempo de inactividad de la aplicación, degradación del desempeño, incompatibilidades técnicas, y la pérdida o corrupción de datos) mediante un cuidadoso estudio previo. El plan de migración define que datos se moverán, a donde se moverán, como se moverán, cuando se moverán y aproximadamente que tanto tiempo tomara el proceso.

Las principales etapas a cubrir en un proceso de migración de sistemas a nivel general son las siguientes;

- Análisis de Impactos en la organización
- Descubrimiento
- Diseño y Mapeo
- Creación de un plan de migración
- Previsiones
- Pruebas antes de la migración
- Migración y Cutover
- Validación de la migración

### *1.4.1 Análisis de Impactos en la organización*

El objetivo de este análisis es identificar los requerimientos operacionales y de negocio que impactan en el proceso de migración. Varios inversionistas o dirigentes dentro de la organización necesitan ser consultados para asegurar que esos requerimientos son contemplados en el plan de migración.

El personal del departamento de tecnologías define el ancho de banda disponible, la carga de CPU, el tiempo de inactividad disponible, y el programa de migración

Los administradores de sistemas y de bases de datos definen los requerimientos de la aplicación y de las bases de datos

Los grupos de seguridad definen los requerimientos de seguridad.

Los dirigentes definen la importancia de y los requerimientos para aplicaciones específicas y tipos de datos.

### *1.4.2 Descubrimiento*

Los detalles sobre la migración de hardware y el ambiente del software son recolectados en esta etapa. El ambiente de migración determina los datos específicos que tienen que ser migrados, por ejemplo la carga de trabajo de operaciones de entrada y salida en los servidores, número de archivos, directorios o volúmenes envueltos en la migración, permisos compartidos, configuración de la red, información del dominio en los servidores DNS.

Un descubrimiento exacto y completo de estos es crítico para el éxito de la migración, y puede ser de tipo manual (mediante línea de comandos) o automatizado (mediante alguna herramienta) el cual se recomienda como opción de preferencia.

No solo el descubrimiento automatizado reduce el tiempo de descubrimiento sino que también reduce la posibilidad de error. Desafortunadamente ninguna herramienta de descubrimiento proporciona todos los datos requeridos así que múltiples herramientas deben de ser usadas aunque se recomienda el método manual para reducir el margen de error.

### 1.4.3 Diseño y mapeo

Durante la etapa de descubrimiento la fuente de datos es identificada, durante el mapeo y diseño el destino de la información es identificada. Hay dos distribuciones básicas de mapeo, uno a uno, donde la distribución fuente y destino son las mismas, y la redistribución donde la distribución fuente y destino son diferentes. Aunque la distribución uno a uno habilita una migración mucho más simple, la migración es a menudo vista como una oportunidad óptima para consolidar y/u optimizar el desempeño y/o capacidades de utilización y la redistribución es un escenario común.

### 1.4.4 Creación de un plan de migración

El plan de migración, el cual es el fin entregable de la fase de planeación funciona como el plano de la implementación de la migración, describiendo las especificaciones del cliente, definiendo los entregables del proyecto y para identificar las metodologías de migración a ser usadas. Existen cuatro entradas principales en el plan de migración.

- Los requerimientos operacionales y de negocios, el cual proporciona las restricciones.
- Datos a ser migrados, con todos los atributos asociados.
- Herramientas de migración disponibles.
- Mejores prácticas en el almacenaje de datos y en la aplicación.

La creación de un plan de migración es a menudo un reto. Diferentes tipos de datos pueden requerir diferentes herramientas de migración y estrategias, y requerimientos operacionales y de negocio (la ventana de tiempo donde la aplicación no estará disponible) pueden requerir de formas creativas para la movilización de datos.



Los componentes de un plan de migración deberían incluir los siguientes puntos:

- Estrategias de migración y actividades claves.
- Dependencias.
- Equipo requerido y herramientas de migración.
- Expectativas del cliente (clientes empresariales, requerimientos técnicos y operacionales).
- Plan de pruebas.
- Procedimientos de verificación.
- Riesgos y planes de contingencia.
- Procedimientos de control de cambios.
- Planificación del proyecto.
- Responsabilidades o actividades después de la implementación.
- Criterio para evaluar la finalización de la migración.

El plan de migración no solo es el entregable en la fase de planeación si no que también es un documento dinámico ya que las variables en el ambiente pueden cambiar o la ejecución puede llevar a resultados inesperados, el impacto en el plan de migración también es documentado.

#### *1.4.5 Previsiones*

Durante esta etapa el ambiente de almacenaje destino es preparado para la movilidad de los datos y sistemas. Volúmenes, directorios y demás son asignados, los atributos de seguridad son puestos, los dispositivos compartidos o exportados son creados. La información generada en las etapas de mapeo y diseño hace posible la automatización de este tipo de tareas. Dentro de esta etapa se considera también planes de contingencia y recuperación a un estado anterior en caso de error en alguna de las partes del flujo de datos.

### *1.4.6 Pruebas antes de la migración*

Antes que cualquier dato sea movido es importante que alguna porción del plan de migración (dependiendo del alcance de la situación específica y con el consentimiento del gerente o cliente) deba de ser migrado y probado. Los resultados de la prueba determinan si una modificación en el plan de migración (por ejemplo la línea de tiempo, herramientas de migración a usar, cantidad de datos migradas por sesión) es requerida. Por ejemplo si las pruebas muestran que el tiempo disponible de inactividad de la aplicación pudiera excederse entonces el plan de migración debe de ser revisado. Este tipo de actividades también puede entregar un tiempo aproximado total del proceso de migración.

### *1.4.7 Migración y cutover*

La migración se podría definir como el proceso de análisis, planificación, diseño y ejecución para trasladar datos y/o procesos de cómputo de un sistema informático a otro. El cutover es la parte ejecutiva de este proceso.

La definición del cutover es: Transferencia de un sistema de cómputo, la transferencia sincronizada de datos, funciones, o usuarios hacia nuevas características o equipo nuevo para asegurar continuidad y minimizar la interrupción de un sistema.

Este es el punto dentro del proceso de migración donde los datos origen pueden ser movidos a los dispositivos destino.

Hay muchas opciones para la transferencia de datos, algunos con sus ventajas y desventajas:

- Las utilidades de migración pueden correr en el host, dispositivo de almacenamiento, o en un dispositivo de red.
- Los datos pueden ser transferidos de dos maneras:

- Fuera de la ruta de datos (fuera de línea) al crear una copia de los datos, moverla ya sea por algún medio de transferencia electrónica, y la aplicación de cambios incrementales, logs de redo, y así sucesivamente.
- En línea (en la ruta de datos), al generar un espejo en tiempo real.

Las metodologías de migración óptima pueden variar significativamente basada en muchos factores como son:

- Los sistemas de almacenamiento origen y destino.
- Topología del almacenamiento de red.
- Localización del equipo.
- Aplicaciones específicas y uso de datos.
- Utilidades de migraciones disponibles y soportadas.

Básicamente no existe ninguna forma correcta, se trata sobre la construcción de una solución basada en el conocimiento del entorno, requerimientos de la organización y operacionales y en la experiencia en migración lo que proporciona un conjunto de mejores prácticas probadas.

La fase de planeación se enfoca en el desarrollo de un plan de migración de excelencia para un proyecto de migración específica. El plan de migración sirve como una guía de implementación.

Una vez que los datos han sido movidos todos los clientes deben de ser redireccionados a los dispositivos destino.

#### *1.4.8 Validación de la migración*

Antes de que la migración pueda ser considerada un éxito, un paso crítico permanece, el validar el ambiente de posmigración y confirmar que todas las expectativas han sido cumplidas, como mínimo, los permisos de los archivos, acceso a la red, estructura de directorios, y que las aplicaciones y bases de datos también sean validados, lo cual a menudo se hace en ambientes de prueba.

En resumen, cada una de las etapas debe de considerar lo siguiente;

<b>Etapa</b>	<b>Productos</b>	<b>Requerimientos</b>	<b>Personal Involucrado</b>
Analisis	Requerimientos operacionales y de negocio.	Apoyo de la alta gerencia.	Inversionistas o dirigentes de la organización. Personal de departamento de tecnologías, bases de datos y seguridad.
Descubrimiento	Detalles de Hardware y software, entorno de aplicaciones.El origen de los datos.	Acceso a la descripción tecnica de los equipos y aplicaciones.	Personal de Operación tecnología y usuarios.
Diseño y mapeo	Se determina el destino de los datos.	Acceso a la descripción tecnica de los equipos y aplicaciones.	Personal de Operación Tecnológica.
Creacion de un plan de migracion	Plan de migración (Implementacion con estrategias, dependencias, equipo, plan de pruebas, riesgos, procedimientos de control de cambio).	Los requerimientos operacionales y de negocios. Datos a ser migrados. Herramientas de migración.	Lider del proyecto de migración, directivos del area informatica y/o personal operativo responsable de los elementos a migrar.
Previsiones	El ambiente destino esta preparado. Planes de contingencia.	Los requisitos de almacenamiento finales.	Lider del proyecto de migración, personal operativo responsable de los elementos a migrar.
Pruebas antes de la migracion	Verificacion exitosa de la migración de una porcion de los datos. Datos aproximados de tiempo, esfuerzo y errores posibles.	Configuraciones de las provisiones.	Lider del proyecto de migración, personal operativo responsable de los elementos a migrar.
Migracion y cutover	Transferencia exitosas de datos de su origen a su destino.	Plan de migración, planes de contingencia. Resultados de las pruebas previas.	Lider del proyecto de migración, personal operativo responsable de los elementos a migrar.
Validacion de la migración	Comprobacion de los datos migrados. Validacion completa del nuevo entorno.	Reporte de la migración, número de registros, archivos, usuarios totales a migrar y migrados.	Lider del proyecto de migración, directivos del area informatica y/o personal operativo responsable de los elementos a migrar.

## 1.5 Metodología en tiempo real

La anterior serie de pasos describen de manera general las etapas que deben considerarse y que se recomiendan cubrir dentro de un proceso de migración informática. La principal característica de una migración en tiempo real es la transparencia de utilización para el usuario final, que permite tener un alto grado de disponibilidad del mismo sin que se vean afectadas las actividades que dependen de dicho servicio, ya que en muchas organizaciones es crucial que el servicio no se suspenda o trabaje de forma limitada, este es un requerimiento de nuestro caso.

La siguiente sección expresa de manera mas especifica algunos de estos pasos que nos permite definir una metodología en tiempo real, enfocándose principalmente en la etapa de creación de un plan donde, en base a la información previamente compilada se puede generar un esquema que engloba todas las funciones del sistema y ayuda a analizar las posibles alternativas del redireccionamiento o contención del flujo de datos para enviarlos al nuevo ambiente y a la vez transportar los datos ya procesados, puntos claves para que una migración de este tipo se complete con éxito.

Una vez identificada la necesidad de un proceso de migración y los requerimientos que produce un análisis detallado se procede a la fase de planeación en donde se deben de tomar en cuenta las siguientes acciones dentro de cada una de sus respectivas etapas.

### 1.5.1 Etapa de descubrimiento

- Identificar todos los elementos técnicos involucrados.
- Desglosar cada elemento y su configuración actual.
- Ubicar a cada elemento dentro del proceso general de datos.
- Identificar cuales elementos son los que tienen menor y mayor concurrencia.
- Identificar las configuraciones internas y externas.

### 1.5.1.1 Identificar todos los elementos técnicos involucrados

Esta acción se refiere a recabar información sobre cada uno de los elementos que componen el sistema, tales como los sistemas operativos donde se ejecuta el sistema de información, manejadores de bases de datos, tecnologías de red, dispositivos de almacenamiento de información, protocolos de red utilizados, reglas de firewall, listas de control de accesos, usuarios, roles y permisos de los archivos, contextos de seguridad, hardware actualizado o especializado, etcétera.

### 1.5.1.2 Desglosar cada elemento y su configuración actual

Sobre cada elemento obtener todas sus configuraciones y propiedades que la hacen funcionar en el sistema. Dentro de esta subetapa se recomienda tener un conocimiento a profundidad de cada parte para conocer cual es el grado de flexibilidad ante posibles eventualidades o para facilitar el proceso de migración y el grado de afectación en la totalidad del sistema al efectuar un cambio en algún parámetro en un elemento específico.

### 1.5.1.3 Ubicar a cada elemento dentro del proceso general de datos

Una vez conocida la función de cada elemento dentro del sistema se debe contextualizar cada elemento dentro de la totalidad del sistema, esto para conocer el flujo de datos que sigue el sistema desde su entrada hasta su salida. Es importante conocer las decisiones que toma cada elemento y bajo que circunstancias ocurren y como se pueden manipular estas decisiones para adecuar o extender su funcionamiento en ayuda del proceso de migración. Se recomienda el uso de alguna herramienta que muestre de manera grafica cada elemento para obtener una visión general y panorámica de la totalidad del sistema y los posibles flujos de datos.

#### 1.5.1.4 Identificar cuales elementos son los que tienen menor y mayor concurrencia y menor tasa de cambios.

Este paso es de gran relevancia y utilidad ya que permite, en medida de su importancia detener momentáneamente y en un periodo bajo de tiempo esa parte específica del sistema y migrarla al nuevo ambiente, se requiere que el elemento que le proporciona datos ahora apunte al elemento migrado, y que este a su vez los proporcione a un elemento que se encuentra en la salida de su flujo de datos, realizando con esto una migración de datos de manera progresiva.

#### 1.5.1.5 Identificar las configuraciones internas y externas

Esto nos permite ubicar el alcance de las configuraciones y de las pruebas para cada elemento a migrar que de no contar con privilegios o facilidades para dichas pruebas se recomienda la instalación de servicios locales que sí proporcionen dichas facilidades o de servicios de prueba gratuitos o de bajo costo que existan en Internet o que sean de manera externa.

### 1.5.2 Diseño y mapeo

- Identificar posibles sustitutos para cada elemento en caso de ser necesario.
- Creación de un plan de migración
- Ubicar aquellos elementos que pueda redirigir el flujo de datos.

#### 1.5.2.1 Identificar posibles sustitutos para cada elemento en caso de ser necesario

En esta etapa se evalúa si existe una versión mas reciente del elemento en cuestión, si existen cambios sustanciales en su configuración o en su formato de datos que retrasen su proceso de migración o que afecte el funcionamiento

del sistema de información. También se evalúa el cambio de dicho elemento por otro que ayude a mejorar el rendimiento, costo y manejo de sistema, dicha decisión debe de estar perfectamente sustentada por un análisis técnico profundo y avalado por los directivos de la sección informática de la institución.

### 1.5.2.2 Ubicar aquellos elementos que pueda redirigir el flujo de datos

Esto con la finalidad de conocer los puntos en donde en base a algunas condiciones se puedan tomar algunas decisiones para modificar el flujo de datos y facilitar así el proceso de migración. Este se convertirá en un punto crítico y estratégico ya que tendrá la capacidad de mandar los datos al nuevo sistema o al viejo dependiendo de ciertas variables y características de los datos a procesar.

Este es el punto en donde sucede la migración en tiempo real ya que no hay necesidad de detener el sistema , solo se conmuta el flujo de datos en base a una o varias características de dichos flujos. Cabe resaltar que en es cambio de dirección pueden existir incongruencias con información previamente almacenada y o procesada, para esto se diseña o planea un proceso para eliminar dicha incongruencia, y esto se ejecuta para cada conjunto de datos que tenga alguna característica en común como pueden ser por ejemplo los archivos de un usuario.

Cada proceso que se ejecute para cada conjunto debe de ser independiente del resto, es decir que se deben de procesar de conjunto en conjunto y no todos al mismo tiempo verificando al final de cada conjunto su correcta actualización, una vez hecho esto se procede al siguiente conjunto y así sucesivamente hasta acabar con todos los datos que tengan alguna cohesión en común.

### 1.5.3 Pruebas

- Instalar en el equipo nuevo cada una de las partes.
- Realizar pruebas del funcionamiento de cada una de las partes.



### 1.5.3.1 Instalar en el equipo nuevo cada una de las partes

Una vez identificados cada uno de los elementos y sus correspondientes configuraciones se pasa a la etapa de instalación en el nuevo equipo u ambiente en donde se migrara, configurándolo de manera adecuada al sistema que lo requiere.

### 1.5.3.2 Realizar pruebas del funcionamiento de cada una de las partes

Una vez instalados y configurados los elementos se prueban de manera separada y en su conjunto utilizando para ello datos que sean lo más aproximados a los que maneja el sistema de información original, esto en medida de lo posible y en función de las características del sistema.

## 1.5.4 Migración y cutover

- Migrar aquellos elementos en orden de actividad.
- En la medida de lo posible utilizar o diseñar y programar una herramienta que automatice el proceso.
- Realizar un simulacro de migración con datos que no intervengan con el funcionamiento normal del sistema.

### 1.5.4.1 Migrar elementos en orden de actividad

Si existieron elementos que se pueden migrar sin que repercuta de manera sustancial el desempeño o funcionamiento del sistema se recomienda migrarlos de uno por uno para evitar complicaciones o confusiones en el flujo natural de los datos.

### 1.5.4.2 En la medida de lo posible utilizar o diseñar y programar una herramienta que automatice el proceso

Si las propiedades del sistema lo permiten se recomienda la creación o utilización de una herramienta que ejecute la migración tomando en cuenta las validaciones y configuraciones correspondientes.

Recabar información para configurar correctamente el ambiente en el que la herramienta funcionara.

### 1.5.4.3 Realizar un simulacro de migración

Utilizando datos que no intervengan con el funcionamiento normal del sistema pero que sean lo mas aproximado a los que utiliza el mismo ejecutar la migración con o sin ayuda de las herramientas necesarias y verificando que no existen errores en dicho simulacro.

## 1.5.5 Validación de la migración

- Verificar la totalidad del proceso
- Realizar los últimos cambios si así se requiere.

### 1.5.5.1 Verificar la totalidad del proceso

Comprobar de manera automatizada o con ayuda de reportes por parte de los usuarios la correcta ejecución del proceso de migración.

### 1.5.5.2 Realizar los últimos cambios si así se requiere

Una vez concluido el proceso se procede a realizar las últimas configuraciones de los elementos que así lo requieran.

Cabe recalcar que no es necesario que cada etapa se complete antes de comenzar la siguiente ya que muchas veces existe la posibilidad o necesidad de regresar a una etapa previa para analizar, configurar o conocer con más detalle uno o varios de los componentes involucrados dando como resultado una serie de acciones paralelas en el proceso de migración.

## CAPÍTULO 2

# ELEMENTOS DE OPERACIÓN PARA LA MIGRACIÓN

### 2.1 Problemática

Dentro de los siguientes capítulos se expondrán un caso de estudio para un problema de migración de usuarios de correo del dominio exalumno.unam.mx y su correspondiente información de estos en el cual se ejemplificará la metodología en tiempo real señalada en el capítulo anterior. Primeramente se expondrán la serie de necesidades que motivaron a esta migración, las dificultades y sus respectivas soluciones que se presentaron durante el proceso de migración, para el cual existió la condición de continuidad de servicio, por lo tanto debe ser en tiempo real.

Los principales motivos de esta migración fueron la necesidad de contar con más capacidad de recursos computacionales –ya que éstas resultaban insuficientes para manejar un creciente número de peticiones de consulta, almacenamiento y procesamiento de correo– y a la necesidad de actualización del software utilizado para desarrollar tales funciones.

El funcionamiento de este sistema de correo es importante para la vinculación de exalumnos de la UNAM con actividades y estadísticas que esta institución académica organiza, difunde y recaba para el beneficio de ellos y el de la comunidad universitaria. Sin embargo, la cantidad de alumnos que el sistema soportaba en un inicio fue rebasado por una demanda constante de espacio y cuentas por parte de los egresados, por lo tanto el equipo y software en que se ejecutaba este sistema dejó de ser eficiente ya que su desempeño se vio afectado al recibir un número elevado de peticiones web y de procesamiento de correos, aunado al deterioro natural de un equipo que está en ejecución conti-

nua y que tiene nullos o breves lapsos de reposo hizo que se requiriera de una nueva infraestructura y plataforma de software para poder brindarle a los usuarios existentes y nuevos un servicio eficiente en la recepción y envío de correos mediante una interfaz web también actualizada. El problema de esta transferencia de servicios e información de un esquema obsoleto a uno mejor, radica en el hecho de que se tiene que realizar en tiempo real, en donde la transferencia sea transparente para los usuarios ya que de no ser así, la imagen y confianza que se tiene en el sitio se vería degradado.

En la siguiente tabla se resumen las causas que originaron esta migración.

Causa	Descripción
Desempeño	El desgaste físico del equipo y el número de peticiones hicieron que las funciones del servidor se vieran afectadas de manera negativa.
Rendimiento	La cantidad de tiempo en la atención de peticiones se ve incrementada de manera significativa.
Crecimiento	Capacidad de usuarios y espacio de almacenamiento insuficiente, causas que afecta la disponibilidad del servicio a nuevos usuarios.
Escalabilidad	Al tener acotada la capacidad de procesamiento del servidor la implementación de software nuevo para brindar nuevas funcionalidades en el servicio se ve restringida.
Operación	La administración de usuarios y cuotas de espacio eran bastante laboriosas, se centraliza en un servidor de directorio.

## 2.2 Antecedentes

El problema de migración de usuarios de correo no ha sido nuevo en este departamento, con anterioridad se presentó el caso del dominio de correo. unam.mx.

La migración de este servidor se originó por motivos de desempeño, seguridad, administración, escalabilidad y manejo de cuotas del correo de los usuarios, así como el espacio ocupado por éste, ya que había que depurarse constantemente, lo cual no siempre se hacía de la manera adecuada, pues se borraban todos los correos de los usuarios para garantizar espacio de disco en el sistema.

El servidor de correo se alojaba en un equipo Sun 3500 y utilizaba sendmail como MTA (Mail Transfer Agent), el cual tomaba su base de datos de usuarios del archivo `/etc/passwd`, y se pretendía migrarlo a un cluster de cuatro computadoras con qmail como MTA, LDAP (Lightweight Directory Access Point) como la base de datos de los usuarios, y Courier IMAP (Internet Message Access Protocol) como servidor de correo POP (Postal Office Protocol) e IMAP. Las computadoras eran, en primera instancia, una PC de escritorio, después se procedería a migrarlos a servidores con mayor poder computacional.

Los aspectos a tomar en consideración durante el proceso de migración fueron:

1. Continuidad del servicio
2. Migración de usuarios
3. Migración de correos viejos

Antes de comenzar con las pruebas la máquina o el servicio se detuvieron cerca de dos días para garantizar la continuidad del mismo, por lo que en esta ocasión se decidió que no se interrumpiera el servicio. La estrategia a seguir fue la siguiente:

1. Instalación de las máquinas de almacenamiento (cuatro Linux, cada una calculando más de 4 400 usuarios).
2. Instalación de la máquina de recepción de mails y reenvío de clientes (POP3 e IMAP).
3. Pruebas de cluster con otro dominio de correo en las máquinas de almacenamiento, para ello se usó `tungsteno.servidores.unam.mx`.
4. Se detuvo el sendmail de `correo.unam.mx` y se encendió el qmail en la máquina, revisando que se realizara de manera correcta la distribución de correo a las máquinas de almacenamiento.
5. Se comenzó la migración de viejos correos a las máquinas de almacenamiento.
6. Se apagó el IMAP de `correo.unam.mx` y se redirigió el módulo IMP (Internet Message Program) de Horde a una de las máquinas de almacenamiento.

El tiempo total transcurrido para realizar la migración (desde apagar el sendmail hasta dar por finalizada la migración) incluyó lo siguiente:

1. Fue necesario todo un día para realizar dicha actividad con calma y revisando cualquier detalle.
2. El mismo día en que se instalaron los servidores Linux quedó lista la máquina de distribución.
3. Se realizaron pruebas durante una semana para verificar cualquier detalle en la configuración.
4. Se requirió de 20 minutos para verificar que todo funcionara de manera adecuada.
5. Incluyendo el nuevo procedimiento, transcurrieron entre 2 hrs. en el caso de la máquina más rápida, y 4 horas en el de la más lenta.
6. El último lapso fue de 10 minutos.

## 2.3 Migración de Usuarios

Para este proceso se utilizaron varios programas y scripts para hacer la conversión de datos del `/etc/passwd` a un árbol de directorio LDAP. Básicamente lo que dicho script realiza es la descomposición de los campos de cada registro y la transformación en atributos de nodos que se insertaran en el árbol de LDAP.

## 2.4 Migración de Correos

Al igual que en el caso anterior, se utilizaron scripts de conversión de formatos de correo. En sendmail se utiliza el formato Mbox, el cual es un archivo al que se le concatenan los correos recibidos y del que se borran los correos eliminados, esto afecta el rendimiento al momento de consultar y alterar el buzón, así como la disposición del mismo cuando llegue un correo corrupto. Qmail utiliza el formato Maildir, el cual almacena cada correo en un archivo diferente, lo que

aumenta el desempeño y depuración de correo. El script hecho en Perl lee el archivo en formato Mbox, se encuentra con ayuda de expresiones regulares las cabeceras de cada correo y las separa en archivos distintos. También escanea el home del usuario para buscar carpetas de correo y convertirlas a este formato, y por último verifica los cambios de permisos y dueño.

Una vez realizada esta conversión, se montan los File Systems de manera remota a las nuevas máquinas para poder transferir los correos.

El problema en este caso fue que el servicio se tuvo que dar de baja temporalmente, con lo que se ven afectados tanto los usuarios como la imagen de la institución.

## 2.5 Operación

A continuación se explicarán los componentes y su función dentro del antiguo sistema de correo del servidor exalumno.unam.mx, llamado también neon.servidores.unam.mx, y posteriormente su funcionamiento en conjunto y su relación en el proceso de migración.

### 2.5.1 LDAP

Se trata de una base de datos jerárquica en forma de árbol invertido cuya velocidad de lectura es superior a la de un SDBD (Sistema Manejador de Bases de Datos) convencional como Postgres o MySQL, pero no así en su velocidad de escritura, por lo cual es bastante adecuada para la consulta de datos.

Esta clase de árbol está constituido por ramas que no son más que un conjunto de nodos u hojas y que a su vez contienen todas ellas un conjunto de datos o atributos que pueden ser obligatorios u opcionales y que están definidos bajo archivos llamados esquemas.

Cada nodo del árbol está referenciado mediante un DN (Distinguished Name), el cual está compuesto en primera instancia por el valor de algún atributo específico, después del nombre de la subrama inmediata superior del nodo y así sucesivamente, hasta llegar al nodo raíz, como se muestra en la ilustración.



## Directorio LDAP

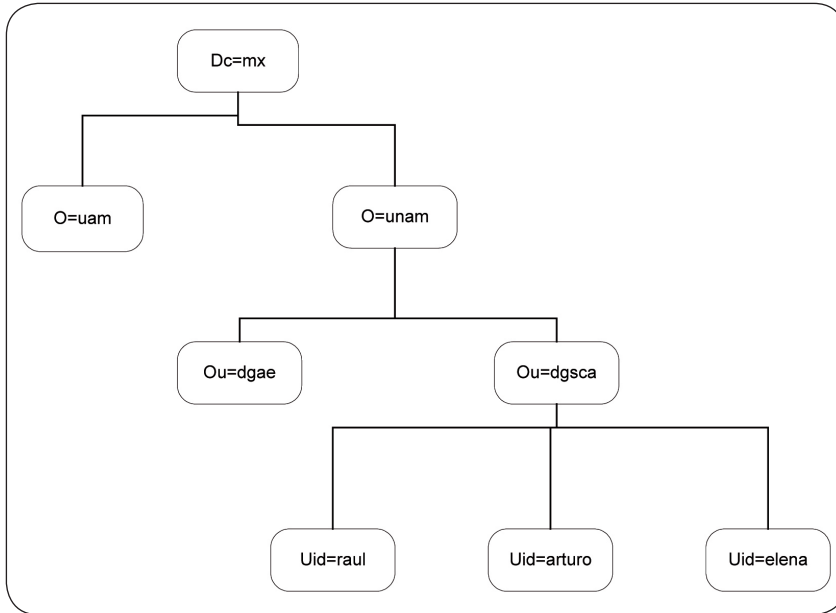


Figura 1. Árbol LDAP

En este ejemplo el nodo principal es dc=mx, lo cual significa que su atributo dc (Domain Component) tiene el valor mx y es a su vez su DN, luego existe una rama con el atributo O (Organization) con valor UNAM por lo que su DN sería O=UNAM,dc=Mx y así sucesivamente para cada rama y nodo del árbol, hasta llegar a las hojas o nodos finales con su atributo uid (User Id) con un valor distinto para cada nodo. De acuerdo con esto, el DN de los nodos finales sería:

```

Dn: uid=raul,ou=dgsca,o=unam,dc=mx
Dn: uid=arturo,ou=dgsca,o=unam,dc=mx
Dn: uid=elena,ou=dgsca,o=unam,dc=mx
  
```

Los nodos son un conjunto de atributos y el DN sólo representa a su identificador dentro del directorio; por ejemplo, la entrada siguiente del árbol dn:uid=raul,

ou=dgsca,o=unam,dc=mx representa al conjunto de atributos del nodo, es decir, esta entrada puede tener más atributos como cn (Country Name), sn (Surname), cn (Common Name). Estos atributos están agrupados en clases que a su vez están definidos en archivos llamados schema, en éstos se definen el tipo de dato, su obligatoriedad y la posibilidad de contener múltiples valores o sólo uno.

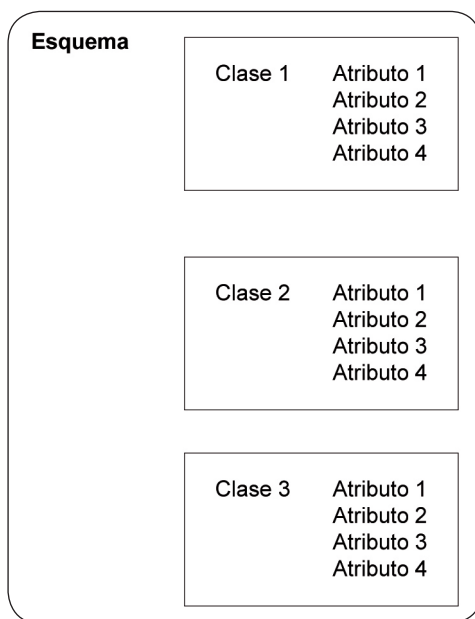


Figura 2. Estructura general de un esquema

Si la instalación de LDAP se hizo mediante rpm's (Red Hat Package Management), estos archivos se encuentran en el directorio `/etc/openldap/schemas`.

### 2.5.2 *DaemonTools*

Es un conjunto de herramientas útiles para administrar y controlar servicios demonios (daemons) en sistemas tipo Unix.

### 2.5.2.1 Características

1. Mayor facilidad para iniciar, detener y reiniciar servicios o procesos.
2. Monitorear procesos.
3. Reinicia automáticamente un proceso cuando éste deja de funcionar.
4. Tiene su propio sistema de bitácoras.

El sistema está basado en el proceso `supervise` que verifica los subdirectorios que se encuentran bajo el directorio `/services` para ejecutar los scripts de nombre `run`, los cuales son los encargados de la ejecución del demonio o servicio en cuestión, tal como lo muestra la siguiente imagen:

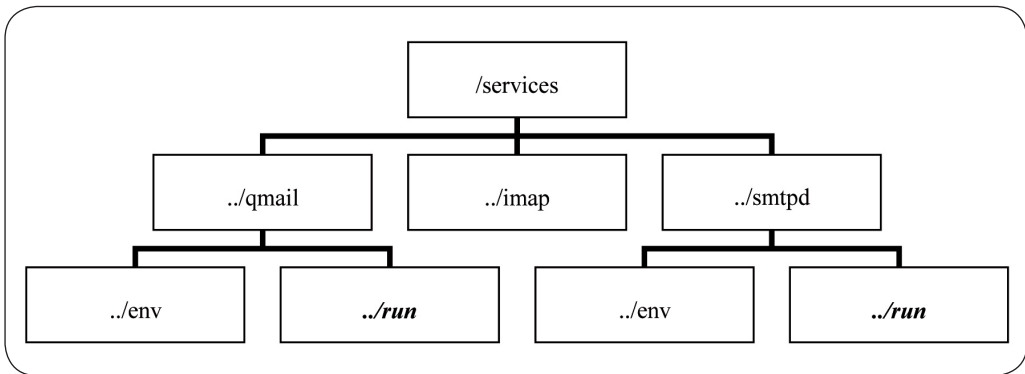


Figura 3. Estructura del directorio `/services`

### 2.5.2.2 Cómo funciona supervise

El proceso `supervise` se cambia a los subdirectorios dentro del directorio `/services` y ejecuta el script `run` si éste existe. Por ejemplo, si existe el directorio `/service/imap`, `supervise` entrará a él e intentará ejecutar el script `run`. Se usa el comando `svc` para levantar y detener el servicio proporcionado por `run`. La información sobre el estado del servicio se maneja con el programa `svstat`.

```
[root@Xalumno ceres]# svstat /service/*
/service/imapd: up (pid 24817) 4747428 seconds
/service/imapdssl: up (pid 24811) 4747428 seconds
/service/pop3d: up (pid 24813) 4747428 seconds
/service/pop3dssl: up (pid 24812) 4747428 seconds
/service/qmail: up (pid 24818) 4747429 seconds
/service/qmqpd: up (pid 24814) 4747429 seconds
/service/qmtpd: up (pid 24815) 4747429 seconds
/service/smtpd: up (pid 24816) 4747429 seconds
[root@Xalumno ceres]#
```

### 2.5.2.3 Script Run y UCSPI (Unix Client-Server Program Interface)

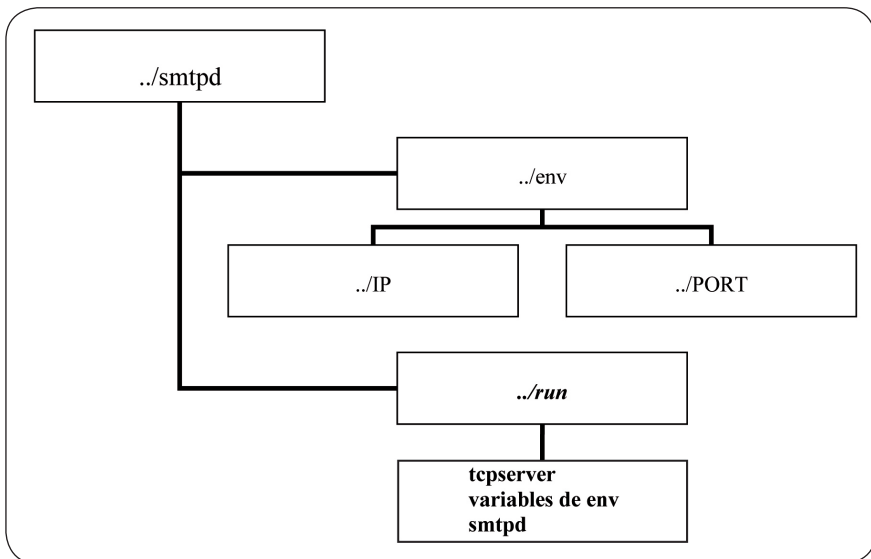


Figura 4. Estructura del directorio donde se aloja el servicio SMTP

Dentro del script run se ejecuta el programa tcpserver, el cual espera por peticiones de red TCP (Transmission Control Protocol). Para cada petición se corre el programa que se le ha dado como parámetro al programa tcpserver. éste puede necesitar algunas variables de ambiente como el nombre del host, el puerto por el cual escuchará, el host que realiza la petición, dirección IP por

la cual escuchará, etc., las cuales toma de los archivos que se encuentran en el directorio `/service/programa_a_levantar/env`. El programa `tcpserver` se usa comúnmente en sustitución del manejador de demonios `xinetd`, ya que éste representaba algunos problemas de Denial of Service.

### 2.5.3 Qmail y SMTP (Simple Mail Transfer Protocol)

Estos servicios se ejecutan bajo scripts run. Para el servicio de SMTP el script run ejecuta el programa `tcpserver` y éste a su vez ejecuta el binario `qmail-smtpd`, con lo cual ya se puede empezar a recibir correo en el servidor. Para el servicio `qmail` existe una pequeña variación, ya que éste no necesita del programa `tcpserver`, es decir, no escucha peticiones de red TCP, ya que su trabajo se limita a procesar la cola de correo que exista en el sistema mediante el programa `qmail-queue` y `qmail-send`, es decir, enviar los correos que los usuarios han enviado a otros sistemas (`qmail-rspawn`) y el de entregar a sus respectivos buzones el correo que llega o que es local (`qmail-lspawn`). El siguiente diagrama muestra el flujo de correo.

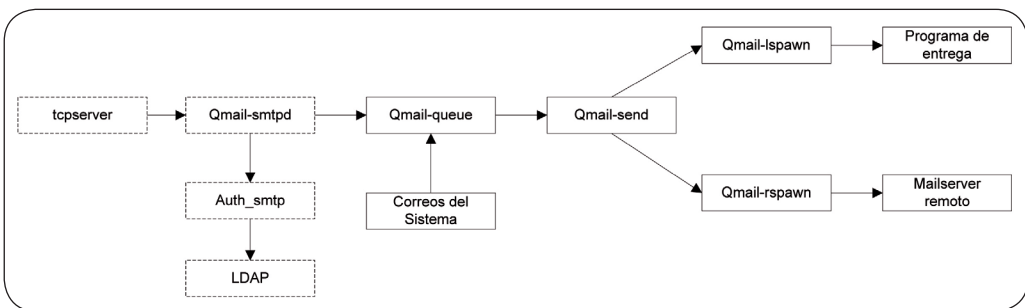


Figura 5. Funcionamiento general de Qmail

### 2.5.4 IMAP Y POP

Es un protocolo de acceso a correos electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier

equipo que tenga una conexión a internet. El servicio de IMAP no se maneja mediante las herramientas DaemonTools en Xalumno, sino mediante un script de inicio en el directorio `/etc/init.d`. Esto cambia cuando se hace la migración al servidor Xalumno donde se utilizan las DaemonTools. Este servicio realiza las mismas funciones que el servicio POP, con la diferencia de que IMAP, al obtener el correo, deja una copia en el servidor. Dichos protocolos son útiles para usuarios que utilizan clientes de correo como Eudora, Outlook y Netscape para recibir todo su buzón de correos en su máquina sin necesidad de bajar de uno en uno.

### 2.5.5 Webmail Horde

Un webmail es una aplicación con interfaz Web hecha con el lenguaje de programación PHP que sirve para consultar y enviar correo con ayuda de los protocolos IMAP y SMTP, respectivamente. En Xalumno se usa una interfaz antigua del webmail Horde. Para esta migración se utilizó una nueva versión que cuenta con módulos para cambio de password, implementación de mensajes de reenvío y de autorespuestas, aunque sólo se utilizó a manera de prueba, ya que para el cambio total se requiere implementar algunas modificaciones al código y autorizaciones administrativas no contempladas en ese momento.

## 2.6. Funcionamiento de Neon antes de la Migración

### 2.6.1 Correo de entrada

Cuando se envía desde internet un correo cuyo dominio sea `exalumno.unam.mx`, la máquina que lo envía verifica el registro MX de este dominio en su servidor DNS (Domain Name Server) para saber a qué dirección IP mandar el correo. En este caso, el MX del dominio `exalumno.unam.mx` era la máquina `neon.servidores.unam.mx`. Una vez enviado el correo, el servicio SMTP de Neon lo recoge, éste ejecuta un proceso llamado `auth_smtp`, el cual consulta la base de datos LDAP para obtener información del usuario destinatario. Los datos que necesita son: el `MailHost` y el `MailMessageStore`, los cuales indican qué servidor recibirá

el correo y en qué directorio lo almacenará, respectivamente. Una vez realizado esto, el correo se pasa al proceso `qmail-queue`, que se encargará de entregar el correo de manera local, tal como ya se explicó en la definición de `qmail`.

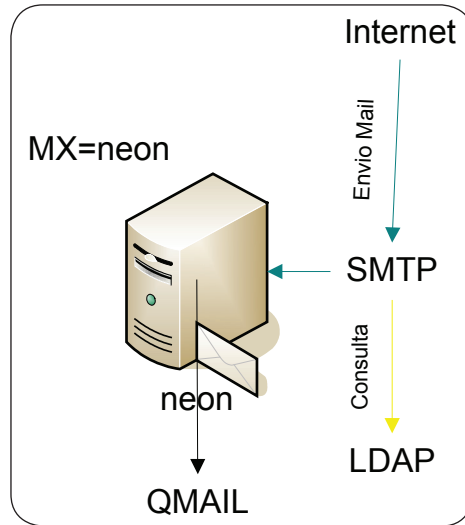


Figura 6. Servicios de Neon

## 2.6.2 Correo de salida

Si se usa algún MUA (Mail User Agent) o cliente de correo como Eudora, Outlook o Thunderbird, el servicio SMTP buscará autenticar al usuario mediante el programa `auth_smtp`. Una vez hecho esto, pasa el correo que se escribió en el cliente al programa `qmail-queue`, el cual se encargará de entregarlo de manera local o remota, dependiendo del dominio. Este tipo de clientes también es útil para consultar y descargar el buzón del usuario en cuestión, esto se logra a través de servicios como IMAP o POP.

## 2.6.3 Operación de Webmail Horde

Horde es un framework hecho en PHP sobre el cual se montan varios módulos útiles para la consulta y manejo vía web de correo. Uno de estos módulos es

IMP, el cual se conecta por medio de una API (Application Programming Interface) en PHP al servidor de IMAP en Neon, que a su vez consulta, con ayuda del programa `auth_imap`, al servidor LDAP para autenticar al usuario y para obtener información del mismo, es decir, en qué host (`mailHost`) y directorio (`mailMessageStore`) se encuentra su buzón para poder mostrarlo a través de la interfaz al usuario ya autenticado. Esta interfaz se conecta también a algún servidor de SMTP para poder enviar el correo de los usuarios, en este caso se conecta al mismo Neon.

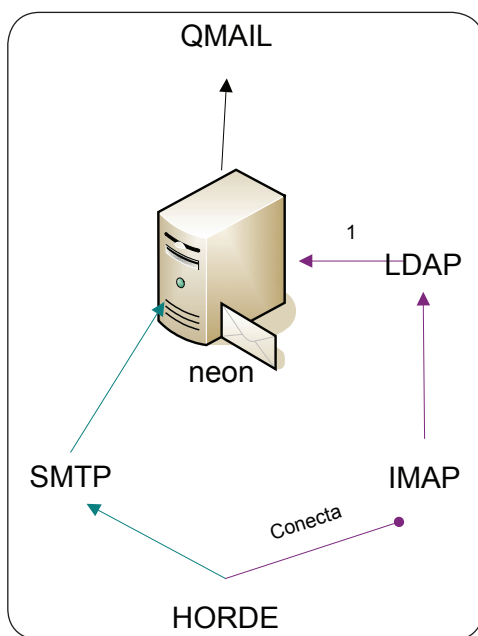


Figura 7. Funcionamiento de Horde en Neon

En resumen, las principales causas por las que fue necesaria esta migración y el porqué de esta manera fueron:

- Aumentar la capacidad de procesamiento y rendimiento de este servicio.
- La necesidad de más espacio en disco duro para los usuarios de correo del dominio `www.exalumno.unam.mx`.



- La actualización del software del servidor de este dominio para evitar fallos de seguridad.
- La necesidad de migrarlo de tal manera fue la de evitar daños a la imagen de la dependencia.
- Una mejor administración de las cuentas de este servicio de correo.
- Actualización y ampliación de las características que ofrecía la interfaz de usuario anterior de este servicio de correo.

## CAPÍTULO 3

# PROCESO DE MIGRACIÓN

### 3.1 Proceso de migración

**P**ara el proceso de migración en tiempo real se requirió que los componentes antes mencionados del servidor Neon interactúen paso por paso con sus correspondientes elementos en el nuevo servidor Xalumno. A continuación se hace una descripción detallada de la planeación del proceso de migración tomando como base a la tabla que resume los pasos de una migración mostrada en el capítulo 1 (punto 1.4).

Primero, fue necesario instalar el sistema operativo en Xalumno. Las características físicas de éste son:

1. Memoria física de 1 Gigabyte.
2. Cuatro procesadores Intel(R) Xeon(TM) a 2.80GHz.
3. Un disco duro SCSI de 35 GB de capacidad.
4. 1 disco duro IDE de 180 GB de capacidad.

### 3.2 Detalles de Xalumno

El sistema utilizado fue Red Hat Advanced Server 3.0 con una instalación base que además incluyó los sistemas manejadores de bases de datos Mysql y PostgreSQL Openldap, tanto cliente como servidor, y el lenguaje de programación PHP con algunos modulos para MySQL y PostgreSQL.

El proceso se desarrolló en el orden que se presenta a continuación:

1. Instalación de las DaemonTools, UCSPI, Qmail e IMAP para el manejo de servicios, de peticiones, entrega de correo y consulta de correo, respectivamente.
2. Hubo que instalar una nueva versión de LDAP en Xalumno por razones de actualización y desuso en varias funciones y atributos del antiguo servidor.
3. Parar por un momento el servicio de LDAP en Neon para poder obtener un respaldo en frío en formato Idif.
4. En Xalumno se corre un script hecho en Perl para poder transformar los atributos que se tenían en el antiguo LDAP al nuevo formato requerido. La salida se guardó en un archivo Idif.
5. Durante el proceso de migración se requirió que en el antiguo servidor no hubiera modificación de la base de datos para evitar incongruencias de la misma entre Neon y Xalumno.
6. Verificar con ayuda del comando Telnet y de sus bitácoras el correcto funcionamiento de los demás servicios.
7. Hacer pruebas con un usuario para garantizar la correcta entrega de correo, esto mediante la inyección de correo local y remota. Para la prueba de correo remota hacia este servidor y sin tener la necesidad de cambiar el registro MX de exalumno.unam.mx, se utilizó un registro DNS temporal y una base de datos LDAP temporal.
8. Crear mecanismos de confianza segura para poder transportar los correos antiguos de Neon a Xalumno mediante el uso de un script en Shell.
9. Comenzar la migración con un script, el cual ejecutara algunos cambios en LDAP y copiara los correos de los usuarios uno por uno.

Después de la instalación del sistema operativo y de los elementos involucrados, hubo que planificar la interacción de esos elementos entre ambos servidores. A continuación se detallan las pruebas entre estos elementos para verificar la correcta comunicación entre ellos.

La primera prueba en la migración fue garantizar la recepción de correo, esto se logró mediante una prueba de inyección de correo en Xalumno (el cual es el nuevo servidor) usando el comando telnet por el puerto de SMTP.

Una vez que se inyecta el correo (mediante el uso de comandos del protocolo SMTP se crea y envía un correo al servidor para su procesamiento) y cuando se han obtenido los datos MailHost y MailMessageStore se presentan dos casos:

- Si el valor del atributo MailHost es igual a Xalumno.servidores.unam.mx el correo se almacenará en el valor que indique el atributo MailMessageStore.
- Si el valor del atributo MailHost es igual a Neon.servidores.unam.mx, el correo será reenviado a este host por SMTP y se almacenará en el valor que indique el atributo MailMessageStore dentro de este host.

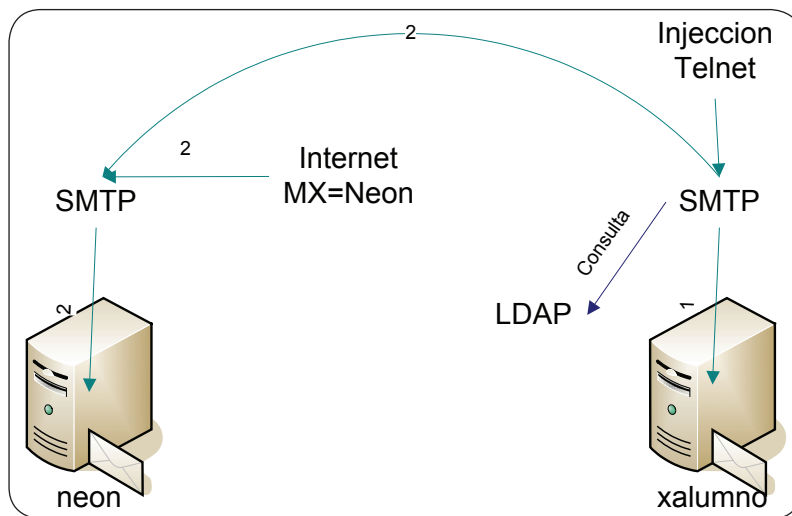


Figura 8. Redireccionamiento de conexiones SMTP

Una vez hecha esta prueba, se hizo el cambio en el registro MX del servidor de DNS, ahora la entrega de correo se hará hacia Xalumno, éste verificará a qué host pertenece el usuario y lo reenviará en caso de ser necesario a esa máquina, es decir, que todo aquel usuario que no tenga como mailHost a Xalumno será redirigido a Neon, su conexión será enviada a éste y así se evita la pérdida de correo mientras se hace la migración usuario por usuario.



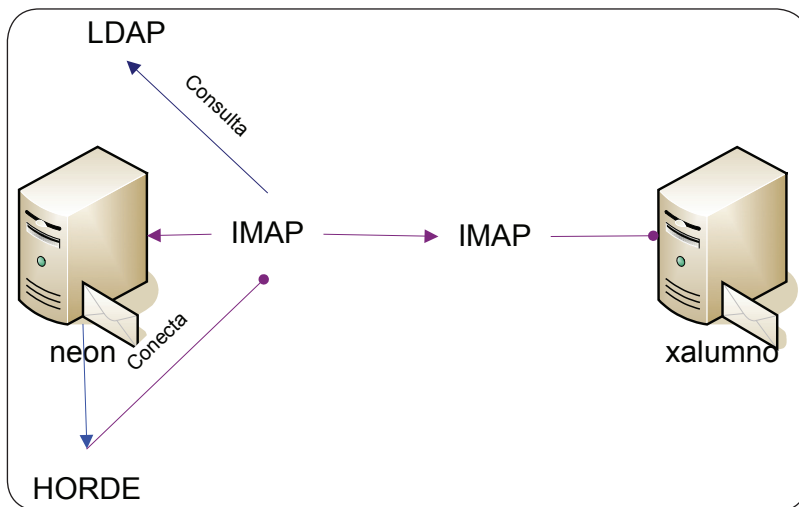


Figura 10. Redireccionamiento de conexiones IMAP

ponibles sólo los nuevos correos, ya que el correo antiguo queda en el maildir del usuario en Neon, por lo que para garantizar que no haya incongruencia entre los nuevos y viejos correos se deben copiar a Xalumno los viejos correos que se encuentran en Neon mediante una transferencia segura usando el programa de conexión remota Secure Shell. Para tal fin es necesario implementar un método de confianza entre ambas máquinas para evitar escribir la contraseña del usuario que realizará la transferencia. El método escogido fue el de llaves públicas y privadas. Todos estos cambios, tanto de atributos como de correos entre ambas máquinas se automatizan en un shell script, y se deben completar para cada usuario y así, si existiera un posible error ya sea de conexión o de lógica en el proceso, no se verían afectados todos los usuarios del sistema al hacer primero un cambio general en LDAP y después de transferencia de archivos, es decir, el cambio se debe hacer usuario por usuario para no modificar el funcionamiento de entrega y consulta de todos los usuarios del sistema.

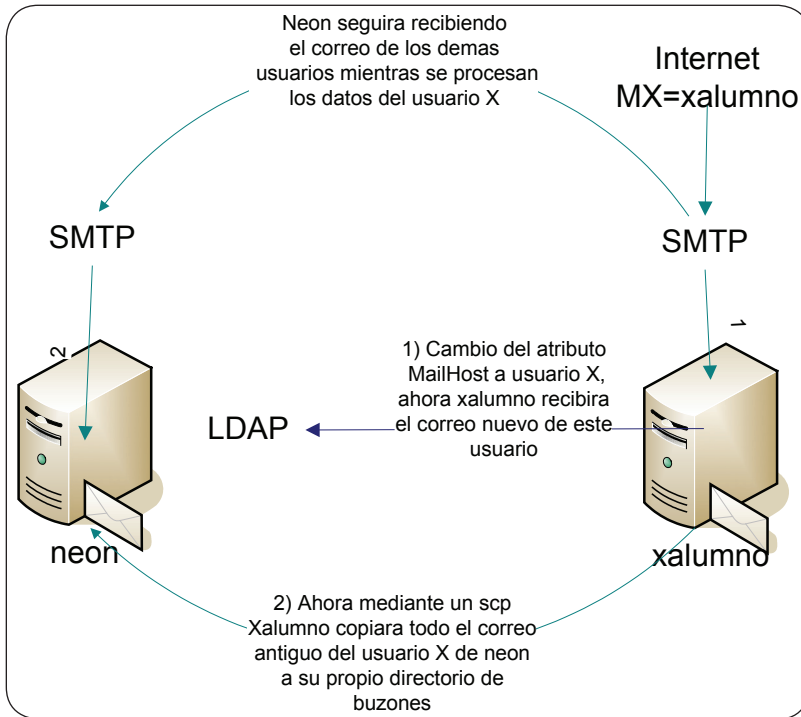


Figura 11. Proceso de migración de correos entre ambos servidores

## CAPÍTULO 4

# CONFIGURACION DE PRUEBAS

### Instalación

A continuación se muestra el procedimiento completo de la instalación y configuración de las herramientas involucradas.

#### 4.1 Instalación Qmail-Ldap

##### 4.1.1 Ventajas

1. Manejo de cuotas con aviso de saturación al 90 por ciento.
2. Almacenamiento distribuido.
3. Base de datos alojada de forma independiente (perfil de usuario).
4. Creación de bitácoras de las actividades de cada usuario.
5. Gran capacidad de almacenamiento.
6. Login y password de 127 caracteres.

##### 4.1.2 Desventajas

1. No existe mucha documentación, por lo cual algunas configuraciones son el resultado del proceso de prueba y error.
2. El sistema es muy modular, por lo que para poder configurarlo correctamente y de acuerdo a las necesidades del sistema hay que investigar cada uno de sus elementos.



## 4.2 Instalación Linux Red Hat AS 3

La instalación del servidor se hizo con los paquetes básicos y con las siguientes particiones:

- / 9.7G
- /usr 24G
- /var 183G

El tamaño de /var se asignó así porque ahí estarán los buzones y la cola de correo del servidor.

## 4.3 LDAP

LDAP es una base de datos en forma de árbol invertido, es decir, tendrá un nodo principal y a partir de éste se irán creando distintas ramas y nodos para los usuarios.

Para ello son necesarios los siguientes rpm's:

- openldap-clients-2.2.13-2
- openldap-2.2.13-2
- openldap-servers-2.2.13-2

Con el propósito de editar archivo de configuración para colocar un comentario en este archivo y en la gran mayoría de archivos de configuración de programas y demonios en sistemas operativos tipo unix, basta con colocar un # al principio de la línea.

### 4.3.1 Configuración

El archivo de configuración de ldap es /etc/openldap/slapd.conf, entre las cuales se indican algunos de los siguientes parámetros.

```
vi /etc/openldap/slapd.conf
```

La directiva *include* indica la ruta de algún archivo schema que el sistema pueda utilizar; se colocan tantos como se requieran. A continuación se muestra el archivo de configuración.

```
-include Ruta_archivo_esquema1.schema  
-include Ruta_archivo_esquema2.schema  
-include Ruta_archivo_esquema3.schema  
...  
...
```

```
database ldbm           #Tipo de base de datos que se uti-  
                        lizará.  
suffix "dc=mx"         #Nodo raiz o si se requieren algunas  
                        ramas adicionales  
rootdn                 "dn=root,dc=mx"   #DN del administra-  
                        dor de ldap  
rootpw password       #password del administrador  
directory /var/ldap   #Directorio donde residen los archivos  
                        que contendrán los datos  
                        #del arbol
```

*#Esquemas utilizados por ldap, algunos dependen de otros como es el caso del #esquema qmail que depende de los esquemas core, cosine y nis.*

*#Este archivo no debe de contener ningún espacio antes del inicio de cada línea.*

```
include /usr/share/openldap/schema/core.schema  
include /usr/share/openldap/schema/cosine.schema  
include /usr/share/openldap/schema/inetorgperson.schema  
include /usr/share/openldap/schema/nis.schema  
include /usr/share/openldap/schema/qmail.schema  
include /etc/openldap/slapd.access.conf
```

Continúa...

Continúa...

```
pidfile          /var/run/Ldap/slapd.pid
argsfile         /var/run/Ldap/slapd.args

modulepath      /usr/lib/openldap
TLSCertificateFile /etc/ssl/openldap/ldap.pem
TLSCertificateKeyFile /etc/ssl/openldap/ldap.pem
TLSCACertificateFile /etc/ssl/openldap/ldap.pem
database        ldbm # soporte dos tipos: Ldbm y berkeleydb
cachesize       10000 # numero de nodos en memoria
dbcachesize     100000
dbnoLocking
dbnosync
suffix          "dc=mx" # raiz del arbol
rootdn          "cn=Siteadmin,dc=mx" # DN del administrador
rootpw         __password #password del administrador
directory       /var/lib/Ldap # debe pertenecer a ldap, en este
directorio se guardan los datos

index mailAlternateAddress,objectClass,mailHost,mail,uid
LogLevel 256
```

En el caso de Xalumno se deben agregar otros atributos para que el framework de Horde pueda realizar algunos cambios para cada usuario.

```
#Listas de control de acceso para permitir modificaciones a los
atributos por parte del usuario ,por ejemplo el cambio de password
attr=userPassword,deliveryMode,mailForwardingAddress,mail
ReplyText
access to attr=userPassword
    by self write
    by anonymous auth
    by dn="cn=Siteadmin,dc=mx" write
    by * none

access to *
    by dn="cn=Siteadmin,dc=mx" write
    by * read
```

Crear archivo slurp.repllog y otorgar permisos.

```
# touch /var/lib/ldap/slurp.repllog
# chmod 644 /var/lib/ldap/slurp.repllog
# chown -R ldap:ldap /var/lib/ldap
# chmod 755 /var/lib/ldap
```

Configurar al cliente para ver a qué host le hará las consultas.

```
# vi /etc/openldap/ldapserver
localhost
```

Iniciar ldap server:

```
# /usr/sbin/slapd -f /etc/openldap/slapd.conf -u ldap -g ldap
```

o mediante los scripts en /etc/init.d.

```
# /etc/init.d/ldap start
# /etc/init.d/ldap stop
```

### 4.3.2 Comandos ldap de administración

Para agregar un nodo al árbol se utiliza:

```
Ldapadd -x -W -D'cn=Siteadmin,dc=unam' -f archivo.ldif
```

Para borrar un nodo:

```
Ldapdelete -x -W -D'cn=Siteadmin,dc=unam' "dn_usuario"
```

Para modificar un nodo:

```
Ldapmodify -x -W -D'cn=Siteadmin,dc=unam' -f archivo.ldif
```

Para realizar una búsqueda:

```
Ldapsearch -x -W -D'cn=Siteadmin,dc=unam' -b'dc=mx' [filtro]
```

Ejemplo:

```
Ldapsearch -x -W -D'cn=Siteadmin,dc=mx' -b'dc=servidor,  
dc=unam,dc=mx' 'uid=accruz' mailHost [atributo(s)]
```

Donde:

*x*= autenticación simple.

*W* = se pedirá password por medio del prompt.

*D'dn'* (distinguished name)= con él se hará la conexión al LDAP (éste debe tener los permisos adecuados para la rama que quiere consultar).

*b*= el punto o base del árbol donde se iniciará la búsqueda.

*f archivo.ldif*= el archivo donde ldap tomará los datos para agregar o modificar algún nodo en el árbol.

*[filtro]* Puede ser de la manera atributo=valor o mediante alguna notación posfija si se quiere hacer una búsqueda más compleja.

*[atributo(s)]* Lista de atributos separados por comas de los atributos que se quieren mostrar por cada nodo.

Éste es un archivo en formato ldif:

```
dn: uid=ceres,dc=eXalumno,dc=unam,dc=mx
uid: ceres
cn: usuario prueba
sn: prueba
objectClass: top
objectClass: person
objectClass: inetOrgPerson
objectClass: qmailuser
userPassword: ceres
mail: ceres@eXalumno.unam.mx
mailQuota: 10000000S,5000000C
mailMessageStore: /home1/maildirs/ceres/
mailHost: Xalumno.servidores.unam.mx
```

En una búsqueda se especifica al menos uno de los anteriores, filtro, lista de atributos o ambos.

```
# mkdir /root/correo
# vi /root/correo/base.ldif
```

Es necesario crear archivo base.ldif, el cual tendrá los nodos y ramas base para generar la estructura de los DN's de los usuarios.

```
dn: dc=mx
objectClass: dcObject
objectClass: organization
o: UNAM
dc: mx

dn: cn=Siteadmin,dc=mx
objectClass: organizationalRole
cn: Siteadmin

dn: dc=unam,dc=mx
objectClass: top
```

Continúa...

Continúa...

```
dn: dc=unam,dc=mx
objectClass: top
objectClass: dcObject
objectClass: organization
o: Nodo UNAM
dc: unam

dn: dc=eXalumno,dc=unam,dc=mx
objectClass: top
objectClass: dcObject
objectClass: organization
o: eXalumno
dc: eXalumno
```

Agregar la base a LDAP.

```
# ldapadd -x -w aeiou -D'cn=Siteadmin,dc=mx' -f base.ldif
```

Verificar la creación del nodo.

```
# ldapsearch -x -W -D'cn=Siteadmin,dc=mx' -b'dc=eXalumno,dc=unam,dc=mx' 'uid=ceres'
```

Para crear un dump de la base en Neon es necesario que ldap esté offline. Para detener el servicio de LDAP el comando es el siguiente:

```
#/etc/init.d/ldap stop
```

Para generar el respaldo en frío:

```
# slapcat -l dump.ldif -f /etc/openldap/slapd.conf
```

Para levantar el servicio:

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

Para crear base de datos en Xalumno, el servicio debe estar offline o agregar una entrada a LDAP.

```
# slapadd -l nuevo.ldif
```

Donde

*l*= especifica el archivo en formato Ldif para los nodos a ingresar.  
*f*= especifica el archivo de configuración de ldap.

## 4.4 Qmail

Los paquetes básicos para que funcione qmail (core) son:

```
ucspi-tcp-0.88.tar.gz // similar a inetd.conf, es un programa  
que atiende peticiones tcp  
daemontools-0.76.tar.gz //administracion de los demonios de qmail  
qmail-1.03.tar.gz //EL servidor de correo
```

Los parches requeridos son:

```
ucspi-tcp-0.88.errno.patch  
ucspi-tcp-ssl-20020705.patch  
qmail-ldap-1.03-20030501.patch  
daemontools-0.76.errno.patch  
djbdns-1.05.errno.patch
```



Primero habrá que instalar Openssl y después copiar todo el directorio `/usr/kerberos/include` a `/usr/include`, posteriormente hay que instalar el servidor de peticiones TCP.

```
# cd /home/software
# tar -zxvf ucspi-tcp-0.88.tar.gz
# gunzip ucspi-tcp-0.88.errno.patch.gz
# gunzip ucspi-tcp-ssl-20020705.patch.gz
# gunzip qmail-Ldap-1.03-20040201.patch.gz
# cd ucspi-tcp-0.88
# patch -p1 < ../ucspi-tcp-0.88.errno.patch
# patch < ../ucspi-tcp-ssl-20020705.patch
# make setup check
```

Después instalar las DaemonTools:

```
# mv daemontools-0.76.tar.gz /usr/local/
# cd /usr/local
# tar -zxvf daemontools-0.76.tar.gz
```

Crear el directorio `admin` en `/usr/local`:

```
# cd admin/daemontools-0.76/src
# patch < /home/software/daemontools-0.76.errno.patch
# cd ..
# package/install
```

Y crear el directorio `/command` con ligas a `/usr/local/bin`.

#### 4.4.1 Instalar Qmail

```
# cd /home/software
# tar -zxvf qmail-1.03.tar.gz
# cd qmail-1.03
# patch -p1 < ../qmail-Ldap-1.03-20040201.patch
```

Se modifica el archivo Makefile para que se compile con algunas características útiles:

```
# vi Makefile
```

Las siguientes líneas del archivo Makefile deben quedar así:

```
LDAPFLAGS=-DQLDAP_CLUSTER -DEXTERNAL_TODO -DDASH_EXT
-DALTQUEUE -DBIGTODO -DSMTPEXECHECK
LDAPLIBS=-L/usr/local/lib -lldap -llber
LDAPINCLUDES=-I/usr/local/include
ZLIB=-Lz
TLS=-DTLS_SMTDP
TLSINCLUDES=-I/usr/local/include
TSLIBS=-L/usr/local/lib -lssl -lcrypto
OPENSSLBIN=openssl
MNW=-DMAKE_NETSCAPE_WORK
MDIRMAKE=-DAUTOMAILDIRMAKE
HDIRMAKE=-DAUTOHOMEDIRMAKE
SHADOWLIBS=-lcrypt
DEBUG=-DDEBUG
```

Al editar el archivo INSTALL.ids, se debe comentar las demás líneas:

```
# vi INSTALL.ids

groupadd nofiles
useradd -g nofiles -d /var/qmail/alias alias
useradd -g nofiles -d /var/qmail qmaild
useradd -g nofiles -d /var/qmail qmaill
useradd -g nofiles -d /var/qmail qmailp
groupadd qmail
useradd -g qmail -d /var/qmail qmailq
useradd -g qmail -d /var/qmail qmailr
useradd -g qmail -d /var/qmail qmails
```

Continúa...

Continuación

```
# mkdir /var/qmail
# sh INSTALL.ids
# make setup check

# groupadd -g 512 vmail
# useradd -g 512 -u 512 -d /home/vmail -m -s /bin/bash vmail
```

Es necesario copiar el archivo “qmail.schema” creado por qmail-ldap-1.03-20040201.patch en el directorio /usr/share/openldap/schema para que LDAP reconozca los atributos de qmail.

```
# cp /home/software/qmail-1.03/qmail.schema /usr/share/
openldap/schema
```

#### 4.4.2 Configuración de Qmail

Crear el archivo /var/qmail/bin/qmailprocmail, éste entregará el correo de manera local,

```
# vi /var/qmail/bin/qmailprocmail
```

el cual tendrá el siguiente contenido:

```
#!/bin/sh
export MAILDIR=${PWD}/
export TMPHOMEDIR=${MAILDIR}
export HOME=${MAILDIR}
if [ ! -f ${MAILDIR}.mailfilter ]
then
/bin/touch ${MAILDIR}.mailfilter && exit 0
# check if maildrop returned EX_TEMPFAIL (75)
[ $? = 75 ] && exit 111
```

Continúa...

Continuación

```
# otherwise return a permanent error
  exit 100else
  :
  fi
  /var/qmail/bin/preline /usr/bin/maildrop && exit 0

# check if maildrop returned EX_TEMPFAIL (75)
[ $? = 75 ] && exit 111
# otherwise return a permanent error
  exit 100
```

al cual hay que cambiarle sus permisos de accesos:

```
# chmod 755 /var/qmail/bin/qmailprocmail
```

Configurar qmail-conf.

```
# cd /home/software
# tar -zxvf qmail-conf-0.60.tar.gz
# cd qmail-conf-0.60
# gzip -dc ../djbdns-1.05.tar.gz | tar xvf -
# cd djbdns-1.05
# patch -p1 < /home/software/djbdns-1.05.errno.patch
# cd ..
# make -f Makefile.ini djbdns=./djbdns-1.05
# make
# make setup check
```

Incluir el directorio bin de qmail en el path de root.

```
# PATH=$PATH:/var/qmail/bin
```

Registrar el servicio en svscan (lo arranca en cinco segundos).

```
# ln -s /var/qmail/bin/qmail-*-conf /usr/sbin
```

Configurar qmail delivery service.

```
# cd /home/software/qmail-conf-0.60
```

Crear directorio del servicio.

```
# qmail-delivery-conf qmail /var/qmail/service/qmail
```

Registrar el servicio en svscan (lo arranca en cinco segundos).

```
# ln -s /var/qmail/service/qmail /service
```

*Nota:* para que funcione el demonio de qmail (qmaild) se deben poner los archivos de configuración en /var/qmail/control.

Verificar estado del servicio.

```
# svstat /service/qmail /service/qmail/Log
/service/qmail: up (pid 6345) 6 seconds
/service/qmail/Log: up (pid 6346) 5 seconds
```

Para detener el servicio:

```
# svc -d /service/qmail
```

Para levantar el servicio:

```
# svc -u /service/qmail
```

Para modo alarma:

```
# svc -a /service/qmail
```

Es utilizado cuando la cola de correo es muy grande. Acorta el tiempo de vida de un correo en la cola, aplica sólo en /service/qmail. También hace un corte en las bitácoras.

Si no arranca el servicio de qmail, será necesario verificar que existan estos archivos y que contengan las líneas siguientes:

```
# vi /service/qmail/run
#!/bin/sh
exec /var/qmail/rc
```

Crear el archivo rc:

```
# vi /var/qmail/rc
#!/bin/sh
exec env - PATH="/var/qmail/bin:$PATH" \
qmail-start ./Maildir/ splogger qmail
#si splogger esta presente lo mandara a las bitacoras de
syslog, si no
#al archivo por default
# chmod 700 /var/qmail/rc
```

Verificar los mensajes en log del servicio:

```
# tail -f /service/qmail/Log/main/current
```

### 4.4.3 Configurar qmail SMTP service

```
# cd /home/software/qmail-conf-0.60
#vi /var/qmail/control/me
xalumno.servidores.unam.mx
```

Crear el directorio del servicio:

```
# qmail-smtpd-conf qmaild qmail /var/qmail/service/smtpd
```

Editar archivo tcp:

```
# cd /var/qmail/service/smtpd
# vi tcp
    127.0.0.1:allow,RELAYCLIENT=""
    132.248.120.:allow,RELAYCLIENT=""
# make
```

Este archivo es el control de acceso para el servicio de SMTP. También se pueden establecer políticas o características adicionales para este servicio, por ejemplo, permitir relay a ciertos host o a usuarios que previamente se hayan identificado para redes o máquinas específicas.

Para agregar más relayclient:

```
# ./add-relayclient 132.248.10.4 132.248.115.
```

Se puede modificar el usuario (qmaild) con el cual se ejecuta el script de inicio /var/qmail/service/qmqpd/run. Aplica para los servicios SMTPD, pop3d, pop3dssl, imap, imaps, qmqpd, y qmtpd.

```
# vi /var/qmail/service/smtpd/run
envuidgid qmaild \
```

Registrar el servicio en svscan (lo arranca en cinco segundos):

```
# ln -s /var/qmail/service/smtpd /service
```

Verificar estado del servicio:

```
# svstat /service/smtpd /service/smtpd/Log
/service/smtpd: up (pid 6345) 6 seconds
/service/smtpd/Log: up (pid 6346) 5 seconds
```

Para probar smtp a mano (primero se deben configurar los archivos en /var/qmail/control), se procede de la siguiente manera:

```
# telnet 127.0.0.1 25
MAIL FROM: jrauda@servidor.unam.mx
RCPT TO: ceres@exalumno.unam.mx
DATA
SUBJECT: hola
HoLa
.
quit
```

Verificar los mensajes en log del servicio con

```
# tail -f /service/qmaild/Log/main/current
```

## 4.5 Instalación de Courier-IMAP

Como usuario root:

```
#chmod -R 755 /usr/src/redhat
#cp courier-imap-3.0.3.tar.bz2 /usr/src/redhat/SOURCES/
```

Continúa...



Continuación

```
#cd /usr/src/redhat/SOURCES/  
#tar xvjf courier-imap-3.0.3.tar.bz  
#cp /usr/src/redhat/SOURCES/courier-imap-3.0.3/courier-imap.spec  
/usr/src/redhat/SPECS/courier-imap.spec  
#rm -rf /usr/src/redhat/SOURCES/courier-imap-3.0.3
```

Como usuario sin privilegios:

```
#su - otroUsuario  
#c /usr/src/redhat/SPECS/  
#rpmbuild -bb courier-imap.spec  
#Logout
```

Como root:

```
#cd /usr/src/redhat/RPMS/1386/
```

Configurar qmail POP3 service:

```
# cd /home/software/qmail-conf-0.60
```

Crear el directorio del servicio:

```
# qmail-pop3d-conf qmaild qmail /var/qmail/service/pop3d
```

Editar archivo tcp:

```
# cd /var/qmail/service/pop3d  
# vi tcp  
132.248.120.:allow
```

Para agregar más clientes:

```
# ./add-client 132.248.10.4 132.248.115.  
# make
```

Registrar el servicio en svscan (lo arranca en cinco segundos):

```
# ln -s /var/qmail/service/pop3d /service
```

Verificar estado del servicio:

```
# svstat /service/pop3d /service/pop3d/log  
/service/pop3d: up (pid 6345) 6 seconds  
/service/pop3d/log: up (pid 6346) 5 seconds
```

Probar conexión local de POP:

```
# telnet 127.0.0.1 110  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
+OK <pid.timestamp@your.server.host>  
user ceres  
+OK  
pass ceres  
+OK  
quit  
+OK  
Connection closed by foreign host
```

Editar archivos de configuración:

```
# vi /service/pop3d/run
```

El cual contiene las siguientes instrucciones:

```
#!/bin/sh
exec 2>&1 \
envdir ./env \
sh -c '
    case "$REMOTENAME" in h) H=;; p) H=p;; *) H=H;; esac
case "$REMOTEINFO" in r) R=;; [0-9]*) R="t$REMOTEINFO";; *)
R=R;;    esac
exec \
softlimit ${DATALIMIT+"-d$DATALIMIT"} \
/usr/local/bin/tcpserver \
    -vD"$H$R" \
    ${LOCALNAME+"-l$LOCALNAME"} \
    ${BACKLOG+"-b$BACKLOG"} \
    ${CONCURRENCY+"-c$CONCURRENCY"} \
    -xtcp.cdb \
    -- "${IP-0}" "${PORT-110}" \
/var/qmail/bin/qmail-popup "${POPUPHOST-`sed 1q /var/qmail/
control/me`} " \
    /var/qmail/bin/auth_pop \
    /var/qmail/bin/qmail-pop3d "${MAILDIRNAME-Maildir}"
'
```

```
# vi /service/pop3d/env/CONCURRENCY
1000
```

Indica el número máximo de conexiones concurrentes del servicio pop3:

```
# vi /service/pop3d/env/IP
0
```

Especifica la IP por la que escuchará el servicio, 0 aplica para todas las interfaces.

```
# vi /service/pop3d/env/LOGLEVEL
2
```

Nivel de mensajes en las bitácoras 0 = sin log, 1 = errores fatales:

```
# vi /service/pop3d/Log/env/MAXFILESIZE  
100kcd
```

Tamaño máximo de archivo de log en Kilobytes:

```
# vi /service/pop3d/Log/env/MAXLOGSIZES  
5
```

Número máximo de archivos de log (por default maneja cinco archivos).  
Es necesario reiniciar el servicio cada vez que se modifiquen las variables de ambiente.

```
# svc -t /service/pop3d /service/pop3d/Log
```

Verificar los mensajes en log:

```
# tail -f /service/pop3d/Log/main/current
```

Configurar qmail IMAP service:

```
# cd /var/qmail/service  
# cp -rp pop3d imapd
```

Es necesario que se conserven los permisos del directorio original, sólo hay que cambiar estas líneas:

```
# cd imapd
# vi run
  -- "${IP-0}" "${PORT-143}" \
  /usr/lib/courier-imap/sbin/imaplogin \
  /var/qmail/bin/auth_imap \
  /usr/lib/courier-imap/bin/imapd "${MAILDIRNAME-Maildir}"
```

Y especificar el puerto por el cual escuchará:

```
# vi env/PORT
      143
# ln -s /var/qmail/service/imapd /service
# svstat /service/imapd /service/imapd/log
```

Verificar los mensajes en log:

```
# tail -f /service/imapd/log/main/current
```

## 4.6 Configurar servicios SSL

Para que funcionen los servicios con SSL es necesario crear el certificado:

```
# cd /home/software/qmail-1.03
# make cert
```

El certificado lo pone en /var/qmail/control y pedirá los siguientes datos:

```
country name: MX
state name: Mexico
locality name: DF
```

Continúa...

Continuación

```
organization name: UNAM
organizational unit name: DGSCA
common name: Administracion de Servidores
email address: staff@servidores.unam.mx
```

#### 4.6.1 POP3SSL

Para la instalación de POP seguro se hace lo siguiente:

```
# cd /var/qmail/service
# cp -rp pop3d pop3dssl
```

Es necesario que se conserven los permisos del directorio original.

Se cambia la siguiente línea en el archivo run, la cual indica el puerto por el que se escucharán peticiones seguras para POP.

```
# cd pop3dssl
# vi run
    /usr/local/bin/tcpserver -s\
    -- "${IP-0}" "${PORT-995}" \
# vi env/PORT
    995
# cp /var/qmail/control/cert.pem .
# ln -s /var/qmail/service/pop3dssl /service
# svstat /service/pop3dssl /service/pop3dssl/log
```

Para probar la conexión segura de pop3dssl a mano, se procede como sigue:

```
# openssl s_client -connect localhost:995 -state -debug
```

Verificar los mensajes en log:

```
# tail -f /service/pop3dssl/Log/main/current
```

#### 4.6.2 IMAPSSL

Se realizan los mismos pasos para el servidor de POP:

```
# cd /var/qmail/service  
# cp -rp imapd imapdssl
```

Es necesario que se conserven los permisos del directorio original.

Se cambia la siguiente línea en el archivo run, la cual indica el puerto por el que se escucharán peticiones seguras para IMAP:

```
# cd imapdssl  
# vi run  
    /usr/local/bin/tcpserver -s\  
penssl s_client -connect localhost:995 -state -debug  
    -- "${IP-0}" "${PORT-993}" \  
# vi env/PORT  
    993  
# cp /var/qmail/control/cert.pem .  
# ln -s /var/qmail/service/imapdssl /service  
# svstat /service/imapdssl /service/imapdssl/Log
```

Para probar la conexión segura de imapdssl a mano, se procede de la siguiente manera:

```
# openssl s_client -connect localhost:993 -state -debug
```

Verificar los mensajes en log:

```
# tail -f /service/imapdssl/Log/main/current
```

## 4.7 Archivos de configuración del sistema de correo

La documentación al respecto está en el archivo `/home/software/qmail-1.0.3/QLDAPINSTALL`

```
# cd /var/qmail/control
```

```
# vi bouncehost  
exalumno.unam.mx
```

Es el dominio por default para cualquier correo generado automáticamente, por ejemplo, `mailer-daemon@servidor.unam.mx`.

```
# vi concurrencylocal  
100
```

Número de procesos concurrentes para entregar correo local, donde local es el mailhost que se encuentra en el archivo `/var/qmail/control/me`, sólo afecta `/var/qmail/service/qmail`.

```
# vi concurrencyremote  
100
```

Número de procesos concurrentes para entregar correo remoto, donde remoto es el mailhost diferente a lo que se encuentra en el archivo `/var/qmail/control/me`, sólo afecta `/var/qmail/service/qmail`.



```
# vi defaultdelivery
./Maildir/
```

Modo de entrega por default de qmail (también es posible /Mbox tipo sendmail y /Mbox en home de usuario):

```
# vi defaultdomain
exalumno.unam.mx
```

Dominio por default que se agrega a los correos de salida en el campo From: que no contengan la arroba:

```
# vi dirmaker
/var/qmail/dirmaker
```

Indica cuál es el programa que creará la estructura del buzón de cada usuario:

```
# vi /var/qmail/dirmaker
#!/bin/sh
mkdir -m 700 -p $1
# chmod 755 /var/qmail/dirmaker
```

Script para generar home o buzón para cada usuario nuevo, es necesario que tenga permisos de ejecución:

```
# vi ldapbasedn
dc=exalumno,dc=unam,dc=mx
```

Base del árbol de ldap donde los servicios de SMTP, qmail e IMAP buscarán la información de los usuarios. El modo en que qmail podrá leer algunos archi-

vos para cada usuario en los que se pueden incluir algunas reglas de filtrado de correo.

```
# vi ldapdefaultdotmode  
both
```

Los posibles valores son:

Ldaponly: no hace caso de los archivos .qmail.

Dotonly: sólo hace caso a .qmail.

Both: hace caso de ambos archivos.

```
# vi defaultquotasize  
20000000
```

Cuota máxima por default, en bytes.

```
# vi defaultquotacount  
5000000
```

Número máximo de correos por default:

```
# vi ldapdefaultquota  
20000000S,1000C
```

Cuotas por default, es un atributo compuesto de los dos anteriores donde s significa storage y c count.

```
# vi ldapgid  
512
```

Id del grupo unix con el cual va a entregar los correos. 512 es el grupo vmail.

```
# vi Ldapuid  
516
```

Id del usuario unix con el cual va a entregar los correos. 512 es el usuario vmail.

```
# vi LdapLocalDelivery  
0
```

Bandera que indica si se verificarán usuarios en ldap (valor igual a 1) o en passwd.

```
# vi LdapLogin  
cn=Siteadmin,dc=mx
```

Login o nodo del LDAP con el que los programas de qmail buscarán información en el árbol de LDAP.

```
# vi Ldapmessagestore  
/var/qmail/maildirs
```

Home o directorio de buzones por default.

```
# vi LdapobjectClass  
qmailUser
```

Filtro de búsqueda por default para LDAP.

```
# vi Ldappassword  
__password__
```

Password para el usuario de LDAP que usarán todos los programas de qmail para buscar la información de los usuarios.

```
# vi Ldaprebind
1
```

Un valor 0 indica la comprobación de usuarios en ldap; sólo compara nombre de usuario y password, 1 compara nombre de usuario y si existe se loggea como nombre de usuario.

```
# vi Ldapserver
127.0.0.1:389 132.248.115.11:389
```

Servidor de ldap y opcionalmente su puerto (por default es el 369), en el cual se harán las consultas de LDAP, se puede incluir más de un servidor.

```
# vi Ldaptimeout
2
```

Tiempo máximo que espera una respuesta del servidor ldap:

```
# vi Locals
exalumno.unam.mx
```

Lista de dominios que qmail considera como locales:

```
# vi me
exalumno.servidores.unam.mx
```

Nombre de la máquina como está dado de alta en el servidor de DNS:

```
# vi queuelifetime  
172800
```

Tiempo máximo en segundos que permanece un correo en la cola de correo sin que pueda ser entregado, después es descartado:

```
# vi quotawarning  
.....mensaje.....
```

Mensaje de aviso cuando la cuota está próxima a saturarse:

```
# vi rcpthosts  
exalumno.unam.mx
```

Contiene todos los dominios de correo que el servidor aceptará:

```
# vi timeoutremote  
40
```

Tiempo máximo para esperar la respuesta de un servidor remoto al cual se le quiere enviar correo:

```
# vi timeoutsmtpd  
40
```

Tiempo máximo para esperar la respuesta de un servidor remoto el cual quiere entregar un correo.

# CAPÍTULO 5

## IMPLEMENTACION

### Migración

#### 5.1 Migración de LDAP

En primera instancia se debe hacer una migración de la base de datos de LDAP de Neon a Xalumno, esto se puede hacer fácilmente parando el servicio y haciendo un respaldo en frío con el comando slapcat, que da como resultado un archivo ldif. Pero debido al cambio de versiones de LDAP, ya no existe el atributo mailQuota, éste se sustituyó por los atributos mailQuotaSize y mailQuotaCount, por lo cual hubo que utilizar un script en perl para poder calcular el valor de estos nuevos atributos a partir del viejo atributo y generar un nuevo archivo ldif para que sea compatible con esta nueva versión. El atributo mailQuota tenia un valor yyyyS,zzzzC donde yyyyS indica la cuota de correo en bytes y zzzzC indica la cuota de correos que tendrá el usuario, por ejemplo, 10000000S,5000000C indica que tendrá una cuota de 10 Mb o 5 millones de correos, lo que ocurra primero. El atributo mailQuotaSize tendrá sólo la cuota de correo y el atributo mailQuota-Count la cuota de correos. La función del script es descomponer el atributo mailQuota en mailQuotaSize y mailQuotaCount, he aquí un ejemplo:

```
mailQuota: 10000000S,5000000C
```

Quedará:

```
mailQuotaSize: 10000000  
mailQuotaCount: 5000000
```

A continuación, el script de migración:

```
#more migración.pl
#!/usr/bin/perl -w
#Se abre el archivo respaldo.ldif que contiene los registros
de la base de datos
open (LDAP,"respaldo.ldif");

#Se guarda todos los registros en el arreglo @registros
@registros=<LDAP>;

# Se recorre el arreglo para mostrar los datos en pantalla
foreach $linea (@registros)
{
#Verifica si alguna línea empieza con la cadena de caracteres
mailQuota:
    if ($linea =~ /^mailQuota:/)
    {
        #Si es así se obtienen los valores de ese atributo para
descomponerlo e imprimir los nuevos atributos en la salida es-
tandar en vez del viejo atributo.
        my @valores=split(/\ /,$linea);
        my @nuevoscampos=split(/,/,$valores[1]);
        @valorQuota=split(/S/,$nuevoscampos[0]);
        @valorCount=split(/C/,$nuevoscampos[1]);
        print "mailQuotaSize: $valorQuota[0]\n";
        print "mailQuotaCount: $valorCount[0]\n";
    }
#De otra manera solo se imprime la línea sin afectar su con-
tenido
    else
    {
        print $linea;
    }
}
# Cerramos el fichero abierto
close (LDAP);
```

Para realizar la migración de los datos de LDAP se ejecutaron los siguientes comandos en el host `neon.servidores.unam.mx`.

Se detuvo el servicio de LDAP momentáneamente.

```
#!/etc/init.d/ldap stop
```

Se crea el respaldo en frío:

```
#slapcat -l respaldo.ldif
```

Se levanta nuevamente el servicio:

```
#!/etc/init.d/ldap start
```

Se copia el archivo respaldo.ldif al servidor exalumno.servidores.unam.mx:

```
#scp respaldo.ldif root@Xalumno.servidores.unam.mx:
```

Dentro del servidor Xalumno.servidores.unam.mx se ejecuta lo siguiente:

Se corre el script en perl para poder actualizar los atributos con sus valores y se manda la salida a un nuevo archivo ldif:

```
#!/migracion.pl > nuevo.ldif
```

Se detiene el proceso de LDAP:

```
#!/etc/init.d/ldap stop
```

Se crea la base de datos en esta máquina a partir del archivo nuevo.ldif :

```
#slapadd -l nuevo.ldif
```



Se levanta nuevamente el servicio:

```
#!/etc/init.d/ldap start
```

La migración de LDAP ha quedado finalizada.

Crear un directorio donde se alojarán los home de todos los usuarios:

```
# mkdir /var/qmail/maildirs
# chown -R vmail:vmail /var/qmail/maildirs
```

Crear un nuevo usuario para poder realizar las pruebas de entrega y consulta de correo:

```
# vi nuevoUsuario.Ldif
dn: uid=calewiro,dc=exalumno,dc=unam,dc=mx
uid: calewiro
cn: Usuairo de Prueba
sn: Usuario
objectClass: top
objectClass: person
objectClass: inetOrgPerson
objectClass: qmailuser
userPassword: hola123
mail: calewiro@exalumno.unam.mx
mailQuota: 10000000S,5000000C
mailMessageStore: /var/qmail/maildirs/calewiro
mailHost: exalumno.unam.mx

# ldapadd -x -W -D'cn=Siteadmin,dc=mx' -f nuevo2.Ldif
```

El programa pedirá el password del usuario para poder ingresar el nuevo nodo.

Verificar creación de usuario nuevo:

```
# /var/qmail/bin/qmail-ldaplookup -u calewiro
```

Usando el comando telnet utilizando el puerto 25 se hacen pruebas de envío y recepción de correo utilizando para ello los protocolos de SMTP e IMAP.

## 5.2 Envío de correo

Desde una Terminal unix en Xalumno se hace una prueba de inyección de correo hacia el mismo host hacia el usuario de prueba, simulando con esto la entrega de correo remoto hacia un usuario local sin necesidad de cambiar el MX del dominio.

```
[root@xalumno ceres]# telnet 127.0.0.1 25
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
220 Xalumno.servidores.unam.mx ESMTP
MAIL FROM: calewiro@gmail.com
250 ok
RCPT TO: calewiro@exalumno.unam.mx
250 ok
DATA
354 go ahead punk, make my day
SUBJECT: PRUEBA DE ENTREGA
HOLA!"B7
.
250 ok 1208825698 qp 24131 by Xalumno.servidores.unam.mx
quit
221 Xalumno.servidores.unam.mx Goodbye.
Connection closed by foreign host.
```

Después se verifica la creación del buzón y la entrega del correo para el usuario de prueba.

```
[root@xalumno cur]# more /var/qmail/maildir/calewiro/Maildir/
cur/1208825698.24163.Xalumno\,S\=407\:2\,
Return-Path:
Delivered-To: calewiro@exalumno.unam.mx
```

Continúa...

*Continuación*

```
Received: (qmail 24131 invoked from network); 22 Apr 2008 00:54:45
-0000
Received: from unknown ([127.0.0.1])
(envelope-sender )
by Xalumno.servidores.unam.mx (qmail-Ldap-1.03) with SMTP
for ; 22 Apr 2008 00:54:45 -0000
SUBJECT: PRUEBA DE ENTREGA
HOLA!"B7
```

### 5.3 Consulta de correo

Nuevamente con ayuda del comando telnet, pero ahora hacia el puerto 143 de IMAP se hace una consulta del buzón del usuario, éste puede ser ejecutado desde cualquier otra máquina. La prueba de consulta de correo por medio del webmail queda ya establecida debido a que éste usa el protocolo IMAP para sus consultas.

```
[root@Xalumno ceres]# telnet 127.0.0.1 143
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^'.
* OK [CAPABILITY IMAP4rev1 ACL ACL2=UNION] Courier-IMAP ready.
Copyright 1998-2004 Double Precision, Inc. See COPYING for dis-
tribution information.
. Login calewiro hola123
. OK LOGIN Ok.
. select INBOX
* FLAGS (\Draft \Answered \Flagged \Deleted \Seen \Recent)
* OK [PERMANENTFLAGS (\Draft \Answered \Flagged \Deleted \Seen)]
Limited
* 50 EXISTS
* 29 RECENT
* OK [UIDVALIDITY 1161812312] Ok
* OK [MYRIGHTS "acdilrsw"] ACL
```

*Continúa...*

Continuación

```
. OK [READ-WRITE] Ok
. fetch 1 fast
* 1 FETCH (FLAGS (\Seen) INTERNALDATE "28-Nov-2006 15:26:50
-0600" RFC822.SIZE 1362)
. OK FETCH completed.
. Logout
* BYE Courier-IMAP server shutting down
. OK LOGOUT completed
Connection closed by foreign host.
[root@Xalumno ceres]#
```

Una vez confirmados estos pasos, se cambia el atributo mailHost a neon.servidores.unam.mx del usuario calewiro y se realizan las mismas pruebas para el usuario, pero ahora las verificaciones deberán hacerse en el host neon.servidores.unam.mx. Con esto se comprueba el redireccionamiento de los protocolos entre host utilizando únicamente el atributo mailHost.

Modificando este atributo en todos los usuarios se consigue que el servicio de entrega de correo permanezca en línea, pero los correos que se tenían almacenados quedan sin posibilidad de ser consultados por parte de IMAP, ya que su referencia apunta a otro lugar en otro servidor, así que el siguiente paso es copiar el buzón de este usuario de Neon a Xalumno para que el correo de la primera prueba pueda ser ahora consultado junto con el correo de la segunda prueba. Esto se logra con un simple comando scp. Ahora bien, como este tipo de cambios es siempre repetitivo, lo ideal fue ponerlos en un script en Shell para realizar dichos cambios para cada usuario obteniendo antes, con la ayuda de consultas a LDAP y utileras en Unix, una lista de todos los usuarios a procesar.

## 5.4 Migración de correo antiguos mediante llaves SSH

Para realizar la copia de los buzones entre los hosts se usó un mecanismo de confianza que proporciona el servicio de Secure Shell llamado de llaves públicas y que proporciona un cifrado de la información que se está transfiriendo. La creación se detalla a continuación.

A la máquina a la cual nos queremos conectar le llamaremos servidor, en este caso Neon, y la máquina que usamos para conectarnos será cliente, es decir, Xalumno. Al usuario conectado en la máquina le llamaremos cliente userC, y al usuario con el que nos conectaremos al servidor, userS.

1. En la máquina Xalumno, loggeado como el usuario userC, tenemos que generar el par de llaves: pública y privada de la siguiente manera:

```
# ssh-keygen -t rsa
```

*Nota:* rsa es el método de cifrado, aunque también existe das. Al teclear la instrucción pedirá una frase privada. Se deja vacía para que Neon no pida la frase en el momento en que userC se quiera conectar como userS.

2. Se cambian los permisos de la llave privada de modo que sólo el dueño pueda leerla.

```
# chmod -R 7600 .ssh
```

3. Esto generará las llaves por default en el directorio `~userC/.ssh` llamadas `id_rsa` (la llave privada) e `id_rsa.pub` (la llave pública), ahora en el directorio `~userS/.ssh` del usuario userS dentro de Neon hay que concatenar el contenido de la llave pública del usuario userC al archivo `authorized_keys`, es decir, después de pasar la llave pública a Neon y estar logueado como userS, hacer lo siguiente:

```
[root@Neon root]#cat id_rsa.pub>> ~userS/.ssh/authorized_keys
```

## En Neon

En este archivo se guardará el contenido de las llaves públicas que se generarán en otras máquinas cliente y que deseen conectarse a Neon como el usuario userS.

#### 4. ¿Cómo funciona la comunicación?

Cuando el usuario userC en Xalumno intenta conectarse a Neon como userS, Neon manda un mensaje cifrado con la llave pública de userC a Xalumno, el cual debe descifrar con la llave privada de userC y mandársela nuevamente a Neon para verificar que el mensaje original enviado y el mensaje descifrado recibido sean los mismos; si es así, la comunicación se establece sin necesidad de pedir alguna contraseña, ya que las llaves y la ausencia de la frase privada garantizan la autenticación del usuario userC en Neon.

De aquí la importancia de que la llave privada sea únicamente accesada por el usuario si se desea implementar una confiabilidad más a la llave privada.

Se comprueba el mecanismo mediante la ejecución del siguiente comando en Xalumno:

```
[root@Xalumno root]# ssh Neon.servidores.unam.mx uname -n  
Neon
```

Se ejecuta el comando en Neon sin necesidad de autenticarse por medio de algún password y gracias a este mecanismo se puede ejecutar remotamente un comando. Teniendo ya un canal de comunicación autenticado y seguro, se procede a ejecutar el script que hará la migración usuario por usuario.

## 5.5 Script de Migración

El script de migración lee los registros de los usuarios de un archivo que se obtuvo de ejecutar un slapcat (respaldo en frío) de la base de datos LDAP obteniendo así un archivo ldif.

Una vez obtenido este archivo se procesa mediante diversas utilerías de Shell para obtener un archivo cuyos campos en cada registro son:

1. La cadena "dn:".
2. El DN completo y sin espacios.

3. La cadena "maiMessageStore:".
4. El directorio donde se almacena el correo del usuario.

Ejemplo de un registro del archivo usuarios.ldif:

```
dn: uid=icovilla,dc=servidor,dc=unam,dc=mx maiMessageStore:
/home2/maildirs/icovilla
```

El script de migración es el siguiente, se incluyen comentarios para la explicación de cada parte del script, los comentarios empiezan con el símbolo #.

```
#more migracionxalumno.sh
#!/bin/bash
#Archivo de donde obtendra Los datos
archivo=usuarios.ldif
#Numero de registros que tiene el archivo usuarios.ldif
numero=`wc -L $archivo |tr -s " " |cut -f1 -d" "`
#contador de registros
cont=1
#Ciclo donde procesara cada usuario hasta alcanzar el total de
registros
while [ "$cont" -le "$numero" ]; do
    #Se crea el archivo cambiar.ldif para que se hagan los cam-
    bios al usuario
    DN=`sed -n ${cont}p $archivo|cut -f1,2 -d" "`
    DIRMAIL=`sed -n ${cont}p $archivo|cut -f4 -d" "`
    echo $DN >>/var/Log/Migracion.Log
    echo $DN >cambiar.ldif
    echo "changetype: modify" >> cambiar.ldif
    echo "replace: mailHost" >>cambiar.ldif
    echo "mailHost: Xalumno.servidores.unam.mx">> cambiar.ldif
    #Se ejecuta el comando para modificar la entrada del usuario
    usando el archivo previamente creado
    ldapmodify -x -w 'passwordLDAP' -D'cn=Siteadmin,dc=mx' -f
    cambiar.ldif -h 2>>/var/Log/LdapMigracionError.Log
    #Si se ejecuto con éxito el cambio no manda error al bipper de vi-
    gilancia ni detiene el script para que no se modifiquen más usua-
    rios
```

Continúa...

Continuación

```

        if [ $? -ne 0 ]
        then
            echo "Fallo el cambio de $cont"
            echo "error al modificar ldap de $DN" |mail
1558338275@skyams.com
            exit 1
        fi
        echo $DIRMAIL
#Se copia el buzón del usuario que existe en Neon a Xalumno de
manera remota
        scp -R root@Neon.servidores.unam.mx:$DIRMAIL $DIRMAIL
#Si hay error manda un aviso y detiene la ejecución del script
para que no copie los demás buzones
        if [ $? -ne 0 ]
        then
            echo "Fallo la transferencia de correo
d $cont"
            echo "Error en el scp de $DN" |mail
1558338275@skyams.com
            echo "Error en el scp de $DN" |mail staff@ser-
vidores.unam.mx
            exit 1
        fi
#Cambia los permisos del buzón al usuario que maneja el sistema de
correo
        chown -R vmail:vmail $DIRMAIL
        cont=`expr $cont + 1`
    else
        echo "error en $DIRMAIL:::$DN" >> /var/log/Ldap
MigracionError.Log
        cont=`expr $cont + 1`
    fi
done

```

Por último, sólo se ejecuta el script en Xalumno:

```
[root@Xalumno root]#./migracionXalumno.sh
```



En la salida del comando sólo se ve el DN del usuario que se está procesando, si existiera un error el script se detendrá.

Por último, para que la recepción de correo ya no sea en Neon sino en Xalumno, habrá que hacer el cambio del registro MX del dominio de exalumno. unam.mx de Neon a Xalumno en el servidor de DNS. Así, cuando llegue un correo a Xalumno, éste sabrá si debe de aceptar la conexión o redireccionarla a Neon según sea el valor del atributo mailHost del usuario en cuestión, como ya ha quedado demostrado en las pruebas anteriores.

La migración ha quedado concluida dejando sólo en Neon el funcionamiento del webmail, ya que éste pasará todas sus consultas a Xalumno.

## CONCLUSIONES

**A**plizando la metodología propuesta en un escenario de migración de servicio de correo electrónico dentro del departamento de administración de servidores queda demostrada su viabilidad y generalización en otros contextos en donde se requiera un proceso de migración considerando siempre las características propias del entorno y considerando siempre las fases de un proceso de migración

La necesidad de contar con una adecuada metodología de migración en el manejo de sistemas de información resulta crucial para evitar pérdidas monetarias, funcionales o la degradación de la imagen pública de una institución ya sea privada o pública.

Si bien las etapas involucradas en un proceso de migración tienen en sí un grado significativo de importancia es de destacar que la que tiene mayor grado de relevancia es la de la planeación ya que en esta descansan las acciones a ejecutar paso a paso en cada parte del flujo de datos así como también contiene información para un posible plan de contingencia o de recuperación en caso de un fallo no considerado, aspectos que deben ser contemplados por el personal que realiza la migración. No olvidemos que todas estas acciones están fundamentadas en base a los requerimientos propios del sistema y la organización.

Un papel fundamental en el proceso de migración es el conocimiento a profundidad de la estructura y flujo de datos en un sistema de información, que van desde la generación de paquetes de red o peticiones a diferentes servicios, hasta la presentación de datos al usuario o su almacenamiento, así como las especificaciones de sus implementaciones que se obtienen de sus respectivas documentación, ya sea gráfica o textual y que en gran medida es abierta y pública, la potencia del uso de herramientas de software libre y del sistema operativo Linux nos brinda la oportunidad de configurar y alterar su comporta-

miento inicial para distintos propósitos, ya sea funcionales o administrativos, o en el caso que nos compete , la migración de un sistema informático. Esto aunado al uso de una metodología de migración mixta, en paralelo y por partes brindan un marco de referencia para la actualización, integración de funciones, ampliación, y distribución de diferentes sistemas de información como pueden ser el correo electrónico, servicios Web, bases de datos, sistemas de archivos compartidos, y básicamente cualquier servicio de red en el que se pueda modificar su flujo y destino , en el cual se pueden ensamblar todas las partes de su funcionamiento de una manera gradual, controlada y con una recuperación de errores de gran confianza y rapidez.

Es importante y crítico contemplar los planes de contingencia en caso de que ocurra algún error en alguna de las partes del proceso, pudiendo restaurar parcial o completamente el servicio a su estado previo para evitar dañar la información del usuario o empresa, afectando la imagen de esta ante sus clientes. El diseño detallado del como interactuaran cada uno de los elementos y su correspondiente contraparte en el nuevo y viejo esquema y como influyen en otros elementos es después de la decisión de migración la parte básica, primordial y esencial en el éxito de un proceso de migración.

## BIBLIOGRAFÍA

RFC 821 SMTP, páginas 1-6, 13, 17,19-21

RFC 3501 IMAP, páginas 6, 8,10, 11,13, 14, 23, 26, 30, 31, 32, 40, 54

RFC 1939 POP3, páginas 2-5, 13,14

Metodologia de Migracion de Datos:

<http://www.docstoc.com/docs/287735/Data-Migration-Methodology>;

[http://www.dulcian.com/Articles/Overview\\_Data\\_Migration\\_Methodology.htm](http://www.dulcian.com/Articles/Overview_Data_Migration_Methodology.htm)

Documentacion de DNS:

<http://tldp.org/HOWTO/DNS-HOWTO.html>

Herramientas Daemon Tools:

<http://cr.yip.to/daemontools.html>

Documentacion de LDAP:

<http://www.openldap.org/doc/admin24/>

Guia Basica de QMAIL:

[http://es.tldp.org/Tutoriales/GUIA\\_QMAIL/guia\\_qmail.pdf](http://es.tldp.org/Tutoriales/GUIA_QMAIL/guia_qmail.pdf)

Integracion de Qmail con LDAP:

<http://www.nrg4u.com/qmail/the-big-qmail-ldap-picture-20031112.pdf>

<http://www.lifewithqmail.org/ldap/>

Documentacion de Horde:

<http://wiki.horde.org/Doc/Admin?referrer=Doc%2FHome#>

Documentacion de SSH:

<http://laura.celdran.name/2005/05/el-protocolo-ssh-su-funcionamiento-en.html>

<http://b52net.blogspot.com/2004/05/autenticacin-con-clave-pblica-en-ssh.html>

<http://es.kioskea.net/contents/crypto/ssh.php3>

## ANEXO

### Estructura de un correo electrónico

Un correo electrónico está compuesto de tres partes, a saber:

1. El envelope o sobre es usado por el MTA para la entrega. Se especifica por los comandos SMTP:

Mail From: correo@remitente

RCPT TO: correo@destinatario.

2. Headers o cabeceras, las cuales son usadas por los clientes de correo. Cada campo de cabecera contiene un nombre seguido de dos puntos y por ultimo el valor del header.
3. El body o cuerpo del mensaje, el cual es el contenido del mensaje del destinatario al remitente. Cuando se usa el comando DATA del protocolo SMTP las cabeceras son enviadas primero seguido de un espacio en blanco y a continuación el body. Cada línea transferida usando el comando DATA debe de ser de menos de 1000 bytes.

El término contenido o content es muy usado para describir la combinación de headers y body. El contenido es enviado por el cliente usando el comando DATA.

A continuación se expone un ejemplo de las cabeceras de un correo electrónico y su explicación.

```

Resent-from: "CARLOS MENENDEZ GARCIA" <@list.cren.
net:carLosmenendez@GEOCITIES.COM>
Resent-to: carlos
Resent-date: Sun, 15 Dec 1996 22:39:02 +0000
Received: from bitnic.cren.net (bitnic.cren.net
[207.86.28.132]) by list.cren.net (8.7.6/8.6.12)
with SMTP id QAA07781 for ; Sun,15 Dec 1996 16:14:20
-0500 (EST)
Message-Id: <199612152114.QAA07781@list.cren.net>
Received: from BITNIC.CREN.NET by bitnic.cren.net (IBM VM SMTP
V2R2)
with BSMTP id 5214; Sun, 15 Dec 96 16:12:33 EST
Received: from BITNIC.CREN.NET (NJE origin LISTSERV@BITNIC) by
BITNIC.CREN.NET (LMail
V1.2a/1.8a) with BSMTP id 5212; Sun, 15 Dec 1996 16:12:30
-0500
Date: Sun, 15 Dec 1996 16:12:28 -0500
From: BITNET List server at BITNIC (1.8a)
Subject: File: "BITNET SERVERS"
To: carLosmenendez@GEOCITIES.COM
Content-Type: multipart/mixed; boundary="-----geoboundary"

```

1. **Resent-from:** Mensaje reenviado por el usuario especificado. En este caso se reenvió en una red local, por eso la dirección no tiene el formato de internet (*geo*). Como CARLOS MENENDEZ GARCIA se encuentra entre comillas, es un distintivo y no tiene significado funcional.
2. **Resent-to:** Mensaje reenviado al usuario especificado. En el caso contemplado se reenvió dentro de una red local al usuario *carlos*.
3. **Resent-date:** Fecha y hora de reenvío del mensaje. En las cabeceras de los mensajes de correo, el campo de la fecha debe estar en un formato determinado por unas normas muy rígidas (piénsese que tiene que poderse interpretar correctamente por una gran cantidad de sistemas distintos). En el ejemplo, se reenvió el domingo 15 de diciembre de 1996 (los programas utilizan siempre el inglés), a las 22:39:02. A la derecha aparece la expresión +0000, que alude a la constante que hay que añadir a la hora GMT para ver la hora real de reenvío del mensaje. En este caso, no hay que añaa-

dir ninguna hora, pero podría haber sido, por ejemplo, +0200, -0300, o cualquier otro valor.

4. **Received:** Cada vez que el mensaje pasa por un servidor aparece este campo de datos, especificándose el nombre del servidor, su dirección IP (número que se asigna a cada servidor), el programa de correo utilizado, y la fecha y la hora en que se recibió en el servidor.
5. **Message-Id:** Número de identificación del mensaje. Se trata de un número único, que lo distingue de cualquier otro mensaje enviado por la red. En el ejemplo, la identificación del mensaje es *99612152114.QAA07781@list.cren.net*.
6. **Date:** Fecha y hora de envío del mensaje. En el ejemplo, el mensaje se envió el domingo 15 de diciembre de 1996, a las 16:22:25, hora correspondiente a GMT-5, es decir a las 21:22:25 GMT.
7. **From:** Remitente original del mensaje. En el ejemplo, el origen del mensaje es *LISTSERV@BITNIC.CREN.NET*, cuyo alias es *BITNET list server at BITNIC (1.8a)*.
8. **Subject:** Asunto del mensaje. El asunto del mensaje del ejemplo es *File: "BITNET SERVERS"*.
9. **To:** Destinatario del mensaje. El destinatario del mensaje es *carlosmenendez@GEOCITIES.COM*
10. **Content-Type:** Tipo de contenido del mensaje; en realidad, es el formato con el que se envió el mensaje.

## Introducción al Protocolo IMAP

El protocolo *IMAP (Internet Message Access Protocol, Protocolo de Acceso a Mensajes de Internet)* es un método que utilizan las aplicaciones cliente de correo electrónico para obtener acceso a los mensajes almacenados de manera remota.

IMAP es utilizado principalmente por los usuarios que pueden obtener acceso a su correo desde varias máquinas, como mensajes almacenados en una ubicación central y a los que puede acceder cualquier sistema que utilice un cliente de correo IMAP y una conexión con el servidor IMAP remoto. Tam-



bién los usuarios que se conectan a internet o a una red privada a través de una conexión de ancho de banda baja utilizan a menudo el protocolo IMAP, puesto que inicialmente se obtiene sólo la información de cabecera del correo.

Una conexión IMAP consiste en el establecimiento de una conexión de red cliente/servidor, la bienvenida inicial desde el servidor e interacciones cliente/servidor. Estas interacciones cliente/servidor consisten en una orden del cliente, datos del servidor y composición del resultado de la respuesta.

Todas las interacciones transmitidas por el cliente y el servidor están en la forma de líneas, esto es, cadenas que terminan con CRLF.

La orden del cliente comienza una operación. Cada orden del cliente lleva como prefijo un identificador (típicamente una cadena alfanumérica corta, por ejemplo A0001, A0002, etc.) denominada *etiqueta* (tag). El cliente genera una etiqueta diferente para cada orden. En el caso de las pruebas de esta migración, la etiqueta utilizada es un punto, regularmente para empezar una conexión se utiliza el comando Telnet, la IP y el número de puerto por donde se esté brindando dicho servicio. Por ejemplo, la ejecución del siguiente comando arroja:

```
[root@Xalumno ceres]# telnet 127.0.0.1 143
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^'.
* OK [CAPABILITY IMAP4rev1 ACL ACL2=UNION] Courier-IMAP ready.
Copyright 1998-2004 Double Precision, Inc. See COPYING for
distribution information.
```

Esto significa que el servicio está en espera de peticiones, el siguiente paso es autenticarse mediante la etiqueta “.”, el comando login, nombre de usuario y el password del usuario en texto claro.

```
. Login calewiro hoLa123
. OK LOGIN Ok.
```

Esto significa que no hubo errores en la autenticación, ahora se ejecuta el comando select para escoger algún directorio de correos.

```
. select INBOX
* FLAGS (\Draft \Answered \Flagged \Deleted \Seen \Recent)
* OK [PERMANENTFLAGS (\Draft \Answered \Flagged \Deleted \
Seen)] Limited
* 50 EXISTS
* 29 RECENT
* OK [UIDVALIDITY 1161812312] Ok
* OK [MYRIGHTS "acdilrsw"] ACL
. OK [READ-WRITE] Ok
```

Con el comando fetch se muestran algunas cabeceras del correo marcado como 1.

```
. fetch 1 fast
* 1 FETCH (FLAGS (\Seen) INTERNALDATE "28-Nov-2006 15:26:50
-0600" RFC822.SIZE 1362)
. OK FETCH completed
```

Finalmente, con el comando logout y con su respectiva etiqueta rompemos comunicación con el servidor de IMAP.

```
. Logout
* BYE Courier-IMAP server shutting down
. OK LOGOUT completed
```

Existen una serie de comandos que permiten crear folders, consultar el correo, etc., que vienen detalladas en el RFC 3501, pero para una mayor comodidad se recomienda el uso de algún cliente de correo.

## Introducción al protocolo POP

El protocolo *POP* (*Post Office Protocol, Protocolo de oficina de correo*) permite a los clientes de correo electrónico recuperar los mensajes de los servidores

remotos y guardarlos en las máquinas locales. La mayoría de los clientes de correo que utilizan el protocolo POP se configuran automáticamente para eliminar el mensaje del servidor de correo después de transferirlo correctamente al sistema del cliente, aunque esto se puede cambiar.

Para establecer una conexión a un servidor POP, el cliente de correo abre una conexión TCP en el puerto 110 del servidor. Cuando la conexión se ha establecido, el servidor POP envía al cliente POP una invitación y después las dos máquinas se envían entre sí otros comandos y respuestas que se especifican en el protocolo. Como parte de esta comunicación, al cliente POP se le pide que se autentifique en lo que se denomina *Estado de autenticación*, donde el nombre de usuario y la contraseña del usuario se envían al servidor POP; al igual que en el caso de IMAP, existe una serie de comandos que permite manipular el correo y se documenta en el RFC 3206. La prueba de POP se explica a continuación.

Se ejecuta un Telnet al localhost y que accederá al puerto 110:

```
[root@Xalumno cur]# telnet 127.0.0.1 110
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
```

Se logró una respuesta por parte del servidor, el siguiente paso es autenticarse:

```
+OK
user calewiro
+OK
pass hola123
+OK
```

Con esto se comprueba el correcto funcionamiento del servidor POP. Finalmente, con el comando quit abandonamos la sesión con el servidor:

```
quit
+OK
Connection closed by foreign host.
```

## Introducción al protocolo SMTP

SMTP está basado en *la entrega punto-a-punto*; un cliente SMTP contactará con el servidor SMTP del host de destino directamente para entregar el correo. Guardará el correo hasta que se haya copiado con éxito en el receptor. El RFC 821 define un protocolo cliente/servidor. Como siempre, el cliente SMTP es el que inicia la sesión (el emisor) y el servidor el que responde a la solicitud de sesión (el receptor). Sin embargo, como el cliente suele actuar como servidor para un programa de correo del usuario, es más sencillo referirse a él como emisor SMTP, y al servidor como receptor SMTP.

## Intercambio de correo

El diseño de SMTP se basa en el modelo de comunicación mostrado en la siguiente figura:

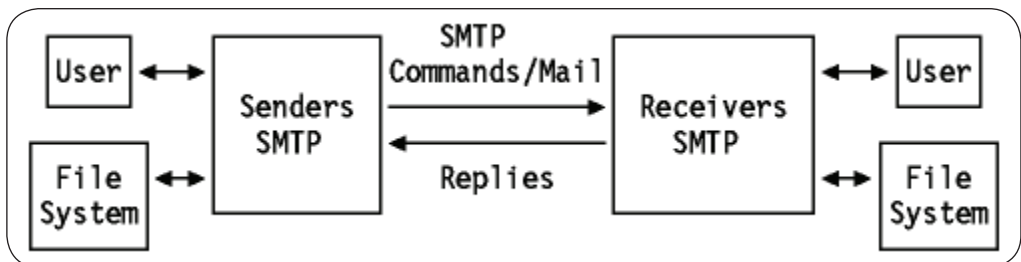


Figura 12. Funcionamiento del protocolo SMTP

Como resultado de la solicitud de correo de un usuario, el emisor SMTP establece una conexión en los dos sentidos con el receptor SMTP. El receptor puede

ser el destinatario final o un intermediario (pasarela de correo). El emisor generará comandos a los que replicará el receptor.

## Flujo de transacción de correo de SMTP

Aunque los comandos y réplicas de correo están definidas rígidamente, el intercambio se puede seguir en

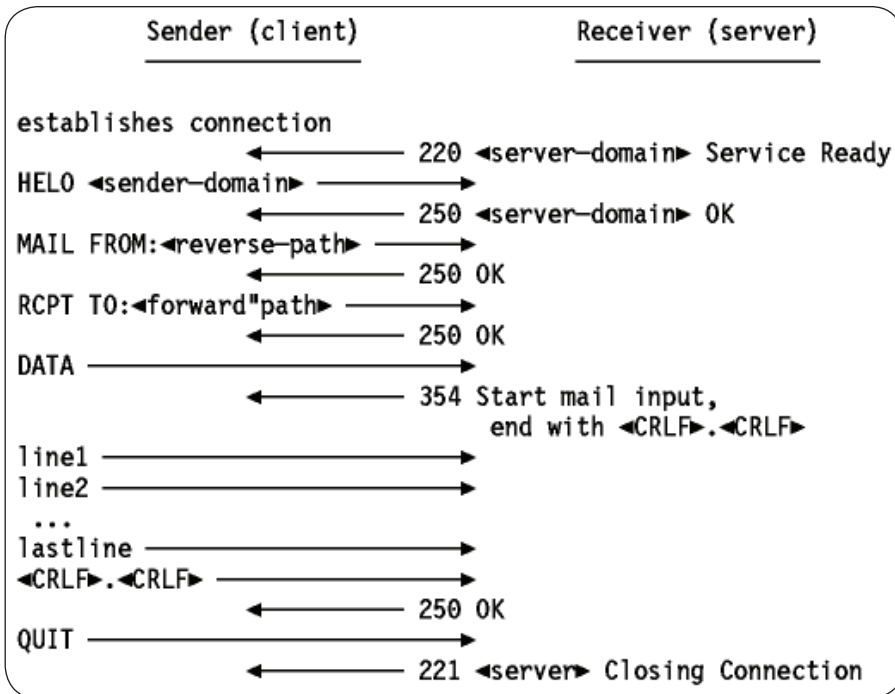


Figura 13. Algunos comandos del protocolo

Todos los comandos, réplicas o datos intercambiados son líneas de texto delimitadas por un <CRLF>. Todas las réplicas tienen un código numérico el comienzo de la línea.

1. El emisor SMTP establece una conexión TCP con el SMTP de destino y espera a que el servidor envíe un mensaje "220 Service ready" o "421 Ser-

*vice not available*” cuando el destinatario es temporalmente incapaz de responder, he aquí un ejemplo realizado en las pruebas:

```
[root@Xalumno ceres]# telnet 127.0.0.1 25
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
220 Xalumno.servidores.unam.mx ESMTP
```

2. El emisor inicia ahora una transacción enviando el comando MAIL al servidor. Este comando contiene la ruta de vuelta al emisor que se puede emplear para informar de errores. Nótese que una ruta puede ser más que el par *buzon@nombre de dominio del host*. Además, puede contener una lista de los hosts de encaminamiento. Si se acepta, el receptor replica con un “250 OK”.

```
MAIL FROM: calewiro@gmail.com
250 ok
```

3. El segundo paso del intercambio real de correo consiste en darle al servidor SMTP el destino del mensaje (puede haber más de un receptor). Esto se hace enviando uno o más comandos RCPT TO:<*forward-path*>. Cada uno de ellos recibirá una respuesta “250 OK” si el servidor conoce el destino, o un “550 No such user here” si no es así.

```
RCPT TO: calewiro@eXalumno.unam.mx
250 ok
```

4. Cuando se envían todos los comandos rcpt, el emisor envía un comando DATA para notificar al receptor que a continuación se envían los contenidos del mensaje. El servidor replica con “354 Start mail input, end with <CRLF>.<CRLF>”. Nótese que se trata de la secuencia de terminación que el emisor debería usar para terminar los datos del mensaje.

```
DATA
354 Start mail input, end with <CRLF>
```

5. El cliente envía los datos línea a línea, acabando con la línea <CRLF>. <CRLF> que el servidor reconoce con "250 OK" o el mensaje de error apropiado si cualquier cosa resultara mal.
6. Ahora hay varias acciones posibles:
  - El emisor no tiene más mensajes que enviar; cerrará la conexión con un comando QUIT, que será respondido con "221 Service closing transmission channel".
  - El emisor no tiene más mensajes que enviar, pero está preparado para recibir mensajes (si los hay) del otro extremo. Mandará el comando TURN. Los dos SMTPs intercambian sus papeles y el emisor que era antes receptor puede enviar ahora mensajes empezando por el paso 3 mencionado con anterioridad.
  - El emisor tiene otro mensaje que enviar, y simplemente vuelve al paso 3 para enviar un nuevo MAIL.

```
SUBJECT: PRUEBA DE ENTREGA
HOLA!"B7
.
250 ok 1208825698 qp 24131 by Xalumno.servidores.unam.mx
quit
221 Xalumno.servidores.unam.mx Goodbye.
Connection closed by foreign host
```

Por último, el SMTP lo pasa a qmail, el cual se encargará de entregar el correo al buzón del destinatario.