UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

"Reingeniería del Servicio de Correo Electrónico en Sistemas Linux"

T E S I S

QUE PARA OBTENER EL TÍTULO DE

Ingeniero en Computación

P R E S E N T A

JOSÉ ALFREDO SÁNCHEZ LÓPEZ

DIRECTOR DE TESIS:
ING. NOÉ CRUZ MARÍN

MÉXICO, D.F Octubre 2008

AGRADECIMIENTOS

A Dios

Por permitirme llegar a este momento.

A mis padres

Por el apoyo que siempre me han brindado. No hubiera sido posible llegar a esta instancia sin su ayuda, me siento muy orgulloso y afortunado por ser su hijo. Los quiero.

A mi familia.

Por todo lo que de ellos he recibido. Quiero incluir a mis hermanos, tíos, primos, etc. No quise poner nombres porque seguro olvidaría alguno.
Una mención especial a mis sobrinos Frida y Osvaldo, quienes llegaron a dar alegría a la casa.

A mis amigos y maestros

Por su compañía, consejos y por su ayuda durante toda mi vida académica y personal.

A Yadi

Por su ayuda incondicional.

También quiero agradecer a Salni (mi perro) por su compañía durante las noches en vela.

ÍNDICE

3. AGENTES DE CORREO ELECTRÓNICO

4. SEGURIDAD

DEFINICIÓN DE LA PROBLEMÁTICA

Los servicios de red tienen tanta utilidad en el mundo actual, que muchas empresas pagan grandes cantidades de dinero para mantenerlos operativos ya que en muchas ocasiones de ello depende la productividad de la empresa.

El correo electrónico es uno de los servicios más utilizados para el intercambio de información, la razón es que ofrece una manera rápida y sencilla de comunicación no interactiva con personas localizadas en casi cualquier parte del mundo. Al principio, el correo brindaba facilidades para el intercambio de información de texto sin formato con lo que la comunicación aunque era muy útil se veía limitada. Sin embargo, debido a la evolución de los sistemas de comunicación y de la tecnología en general, sobre todo la de las redes, el servicio de correo ha evolucionado trayendo consigo varias mejoras. Ahora es posible intercambiar texto con formato, audio, imágenes, vídeo y archivos ejecutables. Estas mejoras han permitido intercambiar libros electrónicos, presentaciones, informes, recordatorios, entre otras cosas, con lo que las empresas aumentan su productividad y se mantienen comunicadas con sus integrantes. Desafortunadamente las bondades que ofrece el sistema de correo se pueden convertir en grandes desventajas si es mal utilizado. La rapidez que proporciona el correo puede ser una forma de propagar virus. Los virus que se distribuyen a través del correo se pueden expandir tan rápido que cuando llegan a detectarse ya infectaron a muchas máquinas alrededor del mundo. La facilidad de enviar muchos correos al mismo tiempo puede consumir grandes recursos de las redes e incluso llegar a saturarlas. Además el correo puede ser un medio para distribuir información racista, violencia o contenido ofensivo.

Existen en el mercado varios servicios de correo gratuitos, gracias a ellos muchas personas se ven beneficiadas al contar con medios de transmisión de correo sin costo, pero son también estos medios en los que proliferan los efectos negativos del correo. Por ello, es necesario contar con sistemas de correo que implementen métodos que traten de evitar situaciones no deseadas en el correo electrónico. En la Facultad de Ingeniería de la UNAM se cuenta con servicio de correo electrónico que tiene como propósito brindar el servicio de correo a la comunidad de la facultad de ingeniería. Los encargados del servicio, conscientes de las amenazas del correo, pretenden renovar el servicio para aumentar la seguridad y privacidad en los mensajes, sin disminuir la rapidez del

servicio, todo ello al menor costo. Para lograrlo se va a utilizar un sistema operativo confiable y seguro como lo es Linux.

OBJETIVO GENERAL

El presente trabajo tiene como principal objetivo el siguiente:

Implantar un sistema de correo electrónico confiable, eficiente y escalable, utilizando mecanismos de seguridad, privacidad y alta disponibilidad que permita el intercambio de contenido diverso en los mensajes. El diseño del sistema debe estar basado en sistemas Linux y apoyado en una metodología de renovación.

El objetivo anterior justifica el título del trabajo “Reingeniería del Servicio de Correo Electrónico en Sistemas Linux”

Adicionalmente al objetivo general se pretende cubrir las demandas de los clientes del sistema (usuarios y administradores)

El sistema debe contar con un diseño que facilite la operación, el mantenimiento y la instalación, que sea portable entre arquitecturas de hardware y portable entre distribuciones de Linux.

CAPÍTULO 1

INTRODUCCIÓN

1.1 ¿QUÉ ES EL CORREO ELECTRÓNICO?

En la actualidad es muy común utilizar el término e-mail o correo electrónico, pero ¿a qué se refiere realmente este concepto que ha revolucionado la comunicación en los medios electrónicos?

El e-mail (Electronic Mail) o correo electrónico; como su nombre lo indica es un medio electrónico para enviar mensajes a través de las distintas redes de telecomunicaciones. Funciona de manera muy parecida a como lo hace el servicio de correo convencional. Un determinado usuario escribe un mensaje a uno o más usuarios y lo envía. Si el destinatario no está presente en el momento de recibir el mensaje, se almacena en lo que se llama buzón electrónico hasta que el usuario pueda leerlo. Esto permite a las personas que se encuentran separadas por una larga distancia física comunicarse de una manera rápida y sencilla, lo que sería muy tardado con el correo ordinario o muy costoso mediante el teléfono

Mediante el servicio de correo electrónico se puede enviar casi cualquier tipo de información ya sea texto, programas o imágenes. En el caso de texto, se codifica normalmente en ASCII, de manera que es posible mandarlo de forma directa. Si se pretende enviar un programa o una imagen, dependiendo de la aplicación que se utilice para enviar el correo, se podrá enviar directamente o habrá que utilizar un programa para convertir el fichero a un cierto formato y decodificarlo en el destino.

El servicio de correo electrónico funciona prácticamente las veinticuatro horas del día, todos los días de la semana. El e-mail no tiene horas de entregas determinadas, más bien es el usuario quien decide cuándo recoger su correo, para hacerlo, sólo necesita tener acceso a su buzón mediante algún dispositivo con acceso a la red. Dicho dispositivo puede ser una computadora, un celular, PDA, etc.

1.1.1 HISTORIA DEL CORREO ELECTRÓNICO

Los antecedentes más cercanos del correo electrónico vienen acompañados con la aparición de las redes de computadoras. En un principio para intercambiar mensajes entre computadoras se colocaban los datos en un directorio conocido, así; cuando se quería saber si había coreos, los usuarios entraban a ese directorio y leían los mensajes, en ese entonces, no había un agente de correo que enviara o recibiera

correo, lo que se hacía simplemente era compartir información mediante el depósito de datos en archivos comunes.

El suceso que sentó las bases del correo electrónico fue realizado por Ray Tomlison en 1971, este ingeniero, ante la problemática del envío de mensajes, creó dos programas para el envío y recepción de mensajes, los programas fueron llamados SNGMSG para enviar los mensajes y READMAIL para recibirlos, éstos programas son los ancestros de lo que hoy conocemos como agente de envío de correo y agente de recepción de correo. La idea de Tomlison era incluir el nombre de usuario y el nombre de la máquina en el mensaje para saber quien era el remitente y el destinatario del mensaje. El separador que se utilizó en aquel entonces fue el signo de la @. Sorprendentemente ese símbolo se sigue usando en el formato estándar actual de las direcciones de correo electrónico.

Una vez que existían los elementos para compartir mensajes, empezaron a tener presencia los programas gestores de correo. En julio de 1971, Larry Roberts desarrolló un programa que permitía la gestión del correo, este programa fue llamado RD, con él se podía leer, archivar, responder y reenviar mensajes así como clasificar los correos por asunto o fecha de envío. Marty Yonke creó un programa llamado *WRD* (BANANARD), este nuevo gestor de correo tenía mejoras respecto al borrado de los mensajes y el entorno del programa era más amigable para el usuario. Con todos los esfuerzos anteriores, John Vittal escribió el primer programa moderno de gestión de correo, el cual incluía todas las opciones de BANANARD, además de direccionar automáticamente las respuestas de correo. Pero sin duda, lo que vino a darle un auge definitivo al correo electrónico fue la gran expansión de Internet, la red de redes hizo del correo electrónico un medio de comunicación a nivel mundial, gracias a él, algunas empresas empezaron a proporcionar cuentas de correo gratuitas, se crearon temas de discusión a través de listas de correo, los académicos compartían ideas mediante mensajes, etc. Lo anterior trajo como consecuencia una gran popularidad en el uso del correo electrónico, tanto así, que el día de hoy es un medio de comunicación electrónica muy importante en la vida cotidiana.

1.1.2 CONCEPTOS DE CORREO ELECTRÓNICO

Mail-boxes

Un mail-box es un archivo o un directorio de archivos donde se depositan los mensajes de correo. Es otras palabras, es el buzón de mensajes de los usuarios que tienen acceso al servicio de correo.

Realmente no existe alguna regla que establezca como es o debe ser el nombre que tenga un buzón de correo. A pesar de ello existen cuentas de correo con un nombre y objetivo especial.

Postmaster

Es una cuenta especial de correo, postmaster es el administrador del sistema de correo, casi todos los programas para transferir correo requieren de una cuenta postmaster. A menudo esta cuenta es un sinónimo de la cuenta real del administrador.

AGENTES DE USUARIO

En el correo electrónico un agente de Usuario, o MUA (Mail User Agent), es un programa que lo ejecuta un usuario para gestionar su correo. Los Agentes de Usuario se utilizan para componer, enviar, revisar, almacenar e imprimir mensajes que se encuentran en el buzón del usuario. Ejemplos de Agentes de usuarios son: elm, mailx, Eudora, Outlook Express, Evolution, etc.

Existe un tipo especial de Agentes de Usuario que se le conoce como webmail. Este Agente se utiliza a través de una página Web y permite realizar las operaciones comunes de gestión de correo para un usuario. El *webmail* es cómodo para mucha gente porque permite ver y almacenar los mensajes desde cualquier sitio con conexión al buzón de correo en lugar de hacerlo desde una computadora específica. La única aplicación necesaria para el usuario es un navegador Web si pretende usar un webmail.

AGENTES DE TRANSPORTE

Los programas que actúan como Agentes de Transporte de Correo (MTA, Mail Transport Agent) se usan para transferir mensajes entre computadoras conectadas a través de una red. Los agentes de usuario pasan el mensaje a un agente de transporte para que éste a su vez, lo entregue a otro u otros agentes de transporte hasta que el mensaje alcance su destino. Los usuarios pueden pasar directamente el mensaje a un agente de transporte sin requerir el uso forzoso de un agente de usuario, sin embargo, a menudo ésta operación sólo la realizan los administradores del sistema, quienes tienen un conocimiento más detallado del funcionamiento del correo y las partes que lo componen.

Los agentes de transporte son los responsables de enrutar los mensajes de forma apropiada hacia su destino. La tarea que realizan los agentes de transporte pasa desapercibida para el usuario, a pesar de que son

ellos los que realizan la tarea más compleja en todo el funcionamiento del correo. MTAs conocidos: Sendmail, qmail, Exim, Microsoft Exchange, Lotus, Postfix, Smail, etc.

AGENTES DE ENTREGA

Los agentes de entrega se usan para depositar los mensajes en los buzones de los usuarios. Cuando un mensaje llega a la máquina destino el agente de transporte cede el mensaje al agente de entrega apropiado, quien será el programa encargado de agregar el mensaje en el buzón del usuario. Algunos agentes de entrega son: mail.local, procmail, maildrop, etc.

LISTAS DE CORREO Y ALIAS

Una lista de correo al igual que un alias representa un conjunto de direcciones de correo. La diferencia entre ambos conceptos radica en que, mientras una lista de correo tiene una cuenta especial que actúa como administrador de la lista el alias no la tiene. Cada elemento de una lista de correo o de una lista de alias puede ser un buzón de un usuario o puede ser otra lista de correo o alias.

CÓDIGO DE CARACTERES

El código de caracteres es un conjunto de bytes que se utilizan para representar los caracteres utilizados en ciertos idiomas. El código más común de caracteres es el US-ASCII el cual consta de 128 caracteres para representar símbolos que se utilizan en el lenguaje inglés. Los 128 caracteres pueden ser fácilmente codificados con 7 bits, por lo tanto el código US-ASCII se le considera un conjunto de caracteres de 7 bits. Muchos lenguajes europeos tienen caracteres acentuados, por ello, utilizan un conjunto de caracteres más amplio que el US-ASCII. La mayoría de los lenguajes que manejan acentos se representan por un conjunto de caracteres de 8 bits, por ejemplo el ISO-8859-1. En el caso de lenguajes asiáticos los cuales manejan muchos caracteres se utilizan conjunto de caracteres multi-byte para poder representar el alto número de símbolos existentes.

CABECERA Y CUERPO DEL MENSAJE

Cada mensaje consiste de dos partes, la cabecera y el cuerpo. La cabecera contiene información acerca del autor del mensaje, el destinatario, la fecha de creación, el asunto del mensaje, etc. Cada elemento de la cabecera tiene una palabra clave que identifica su

función. La mayoría de las palabras clave tiene una traducción directa al español, por ejemplo **Date** se refiere a la fecha, **From** al remitente y así sucesivamente. La separación entre la cabecera y el cuerpo del mensaje generalmente lo representa una línea en blanco. El cuerpo del mensaje contiene la información que el remitente trata de comunicar.

Las líneas más importantes del encabezado son:

- **From:** Es la dirección del remitente. Sólo puede haber una línea de este tipo en el encabezado.
- **To:** Se refiere a uno o más destinatarios del mensaje. Esta línea puede especificar más de una dirección de correo.
- **Cc:** Copia a destinatarios. Ésta línea equivale a la copia en papel carbón en el caso del correo normal.
- **Bcc:** Representa una copia oculta. Cuando se especifica una dirección en este apartado, se manda una copia del mensaje a dicha dirección sin que los otros destinatarios tengan conocimiento de ello.
- **Subject:** Tema del mensaje. El texto que aparece en esta parte del encabezado tiene por objetivo la descripción del contenido del mensaje. Debe ser corto y descriptivo.
- **Date:** Indica la fecha y hora en que el mensaje fue enviado.
- **Message-Id:** Es un identificador de cada mensaje, es único y lo inserta el ordenador que lo envía. Por ejemplo:

  <93116.130423TAMARIRA@EVALUN11.BITNET>

- **Received:** Es la información que se utiliza para comprobar los problemas que hayan aparecido en el reparto de un mensaje. En ella se muestra las direcciones de las máquinas por las que pasó el mensaje en dirección a su destino, junto con la fecha y hora en que lo hizo.
- **Resent-From:** Dirección de la persona o programa desde el cual llega el mensaje. El hecho de decir "reenviado" te notifica de que el mensaje le ha llegado a la persona que se indica en este campo y ella, a su vez, te manda una copia.
- **Reply-To:** Especifica la dirección a la que debes contestar. En ocasiones no es la misma desde donde se ha enviado el mensaje.

PROTOCOLO DE TRANSFERENCIA

El protocolo de transferencia es el conjunto de reglas que controlan la transferencia de mensajes. Este protocolo es el lenguaje que hablan los

Agentes de Transporte para poder enrutar los mensajes. El protocolo de transferencia más conocido es SMTP y es el estándar que se utiliza en la mayoría de los sistemas actuales.

REGISTROS MX

Un registro MX (Mail eXchanger) es una entrada en un servidor DNS que indica la máquina que se encarga de procesar el correo en determinado dominio. Cuando los agentes de transporte necesitan enviar un correo a una red externa, lo primero que hacen es una petición a un servidor DNS para así determinar el servidor de correo del dominio al cual está dirigido el mensaje.

1.1.3 VENTAJAS DEL CORREO ELECTRÓNICO

En la actualidad el correo electrónico es uno de los medios de comunicación más utilizados a lo largo y ancho del mundo. Como todos los medios de comunicación el e-mail tiene ventajas, pero también tiene defectos. Algunas de sus ventajas son, a su vez, desventajas. Por ejemplo, el conjunto de usuarios está de acuerdo en que poder mandar un mismo mensaje a distintas personas al mismo tiempo es una de las más fuertes ventajas del uso del correo electrónico. Sin embargo, esa característica es también, la culpable de la transmisión de muchos mensajes innecesarios y molestos que hacen perder el tiempo al receptor y en casos extremos, sobrecargan las redes de comunicación.

Las ventajas en el correo electrónico se pueden resumir en los siguientes puntos:

FACILIDAD DE USO

Con la aparición de agentes de usuario tan avanzados, el uso del correo se ha incrementado enormemente. Casi cualquier persona puede gestionar su correo de una forma rápida y sencilla.

VELOCIDAD DE ENVÍO

La velocidad con que se transmiten mensajes a través del correo electrónico es muy superior al correo tradicional. El mensaje electrónico pasa de una máquina a otra hasta alcanzar el destino final y ser depositado en el buzón del usuario. El tiempo desde que se escribe el mensaje hasta su recepción puede ser de algunos minutos, incluso si las máquinas están en distintas partes del mundo.

BAJO COSTO

Un correo enviado a través del servicio postal tiene ciertos costos asociados al papel, los sellos, sobres, etc. Para enviar un correo electrónico sólo hace falta tener una cuenta y una conexión a la red. Muchas veces sólo es necesario pagar la conexión a la red, ya que existen sitios donde se puede obtener una cuenta de forma gratuita, además el costo es el mismo si se envía un mensaje a una persona que si se le envía a diez.

ACCESIBILIDAD

Existen numerosos medios que permiten el acceso al correo electrónico. Se puede acceder a él desde una PC, una computadora portátil, un PDA o incluso un teléfono móvil. Por lo tanto los usuarios pueden gestionar su correo casi desde cualquier lugar.

FORMATO

El día de hoy casi todos los programas de correo permiten asignar formato a los mensajes, por ejemplo, se puede usar fuentes de letras diferentes, símbolos gestuales y en algunas ocasiones etiquetas HTML. Todo con la finalidad de darle presentación a los mensajes.

ANEXOS

En un correo electrónico es posible enviar documentos anexos al mensaje, dichos documentos pueden ser texto, hojas de cálculo, imágenes, ejecutables, entre otros archivos. La ventaja es que personas alejadas por una gran distancia física pueden compartir sus documentos de forma rápida. Aunque esa facilidad también ha proporcionado un medio para enviar agentes maliciosos a través del correo.

Obviamente, el correo electrónico no es medio de comunicación perfecto, tiene muchas ventajas, pero también ciertas desventajas que si no se toman en cuenta pueden provocar afectaciones serias. Algunas desventajas en el uso del correo son las que se enlistan:

- Si no se tiene un uso adecuado de una cuenta de correo, algunos usuarios sin permiso podrán leer, borrar o alterar los mensajes.
- Los anexos que se mandan junto con los mensajes pueden contener virus o algún otro elemento no deseado.

- Las cadenas de correo pueden saturar el servidor de correo o incluso saturar la red.
- Existe difamación a través de mensajes de correo.
- Se pueden mandar correos a nombre de otra persona.
- Se cometen engaños y extorsión a través de los mensajes electrónicos.

1.2 REINGENIERÍA

Es muy cierta aquella premisa que clama “lo único constante es el cambio”, y más aún cuando se refiere al cambio en el mundo de la teleinformática. Con la aparición de Internet y las nuevas tecnologías, se ha transformado profundamente no sólo la forma de los medios de comunicación, sino también las formas de vivir. Como consecuencia, han surgido nuevos productos, nuevos servicios y nuevas arquitecturas lo que demanda nuevos profesionales en condiciones de proponer y gestionar el avance tecnológico. El cambio tecnológico trajo consigo un número creciente de usuarios que provoca una reducción de los recursos destinados a los servicios. El mejor ejemplo es la aparición de IPv6 como resultado de la escasez de direcciones IP’s que actualmente se manejan.

Hasta hace algunos años el cambio tecnológico había ocurrido con lentitud lo que daba oportunidad a los operadores de adaptar sus servicios, haciendo ajustes ocasionales o incluso dejando que se acumularan ciertas necesidades de cambio y las heredaban a la siguiente generación. Sin embargo, el día de hoy muchas situaciones no se pueden resolver realizando pequeños ajustes, más bien se necesita un rediseño en el servicio para adaptarse a las nuevas condiciones. Es en este punto es donde hace su aparición la reingeniería, la cual tiene una estrecha relación con el cambio radical y está orientada principalmente a los usuarios.

1.2.1 CONCEPTO DE REINGENIERÍA

Reingeniería es la revisión fundamental y el rediseño radical de procesos para alcanzar mejoras en medidas críticas y actuales de rendimiento, tales como costos, eficiencia en el servicio y rapidez*.

*Hammer & Champy “Reingeniería” (Definición de Reingeniería)

Por lo tanto, la reingeniería pretende alcanzar mejoras en una organización basándose en el rediseño y en la satisfacción del usuario.

A pesar de que existen diferentes formas de alcanzar mejoras en una organización, la conveniencia de cada una de ellas depende de los resultados que se pretendan alcanzar. No siempre será adecuada la aplicación de la reingeniería, sin embargo, existen ciertas situaciones en las que sería bueno considerarla.

Se necesita reingeniería en una organización en ciertas circunstancias:

- El rendimiento de la organización esta por detrás de la competencia.
- Cuando la organización esta en crisis.
- Cuando ciertas condiciones del mercado cambian y es necesario adaptarse a ellas, por ejemplo, la tecnología está en constante cambio.
- Cuando se pretende obtener una posición de líder del mercado.
- Se puede utilizar reingeniería como respuesta a una competencia agresiva.
- Cuando la organización es líder y debe seguir mejorando para mantenerse como tal.

1.2.2 PARTICIPANTES EN LA REINGENIERÍA

Para llevar a cabo la reingeniería se necesitan ciertos elementos como lo son: el líder, el dueño del proceso y el equipo de reingeniería.

El Líder

Es un elemento con visión para reinventar los procesos bajo nuevos esquemas. Necesita tener suficiente jerarquía para persuadir a la gente con la finalidad de que acepten los cambios que la reingeniería implica. El líder designa al dueño del proceso Sin este líder el proceso de reingeniería sólo queda en buenos propósitos sin llegar a culminarse como se espera.

Dueño del Proceso

Es el responsable de un proceso específico. Es importante que el dueño del proceso tenga aceptación de los compañeros con los que va a trabajar ya que de ello depende una buena integración en el proyecto.

Equipo de Reingeniería

El equipo de reingeniería se encarga de producir ideas y planes para rediseñar el proceso. Una parte del equipo debe conocer el proceso actual a fondo y la otra parte debe ser personal ajeno para así proponer alternativas. Un equipo debe trabajar sólo en un proceso al mismo tiempo.

1.2.3 METODOLOGÍA PARA LA REINGENIERÍA

Existen diferentes metodologías en este apartado sólo se describirá de forma breve la metodología rápida Re. Esta metodología se diseñó para que la utilicen equipos de reingeniería en organizaciones sin tener que basarse en expertos ajenos a la organización.

Etapa 1 – Preparación

En esta primera etapa se realiza la búsqueda de metas y se forma al equipo encargado de hacer reingeniería. También se definen los puntos que justifican la ingeniería para determinado proceso.

Etapa 2 – Identificación

En esta etapa se hace un análisis del valor del proceso. Además se desarrolla un modelo orientado al cliente. Para realizar lo anterior se requiere un conocimiento profundo de la organización así como de sus procesos.

Etapa 3 - Visión

El propósito de esta etapa es desarrollar una visión del proceso capaz de producir un avance decisivo en rendimiento. La visión del nuevo proceso debe ser comprensible para todo el personal.

Etapa 4 – Solución

En esta etapa se realiza un diseño técnico y un diseño social basándose en la visión (etapa 3).

Etapa 5 – Transformación

Implementa las visiones del proceso, aplicando el diseño de la etapa de solución.

A pesar de que la reingeniería busca cambios rápidos, es posible complementarla con la mejora continua.

1.3 LINUX

En el mundo de las tecnologías de la información es muy común utilizar el término “Linux”, aparece en revistas, en sitios de Internet, en conferencias, etc. Con todo lo anterior, surge una pregunta obligada, ¿qué es linux?

1.3.1 ¿QUÉ ES LINUX?

Hablando de forma estricta, Linux se refiere al kernel o núcleo de un sistema operativo tipo Unix de libre distribución y que está formado con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos y grupos de software, este sistema operativo recibe el nombre de GNU/Linux. El kernel (lo que realmente es Linux) es el encargado de gestionar recursos a través de servicios de llamada al sistema, en otras palabras, es el organizador, administrador e intermediario entre los componentes hardware del sistema y las aplicaciones de usuario.

Al sistema operativo GNU/Linux se le conoce simplemente como Linux. Por lo tanto, se puede decir que Linux es un sistema operativo basado en Unix con características modernas como lo son: memoria virtual, multitarea real, bibliotecas compartidas, módulos cargados sobre demanda, soporte para TCP/IP, soporte para entornos gráficos, etc. Además, Linux es un sistema operativo libre, lo que significa que se tiene plena libertad para modificar su código fuente y distribuirlo sin tener que afrontar problemas legales.
Las características más relevantes del sistema son:

- Es un sistema multiusuario en tiempo compartido, es decir, un sistema en el que pueden trabajar varios usuarios simultáneamente compartiendo la CPU y todos los demás recursos del sistema. Cada usuario puede ejecutar varios procesos a la vez.

- El sistema operativo está escrito (en su mayoría) en un lenguaje de alto nivel (lenguaje C), lo cual hace que sea fácil de leer, entender, modificar y transportar.

- La interfaz de usuario (shell bash) es sencilla y potente, puede ser reemplazada por otra en cualquier momento si se desea.

- Permite construir programas grandes a partir de otros más sencillos.

- El sistema de archivos tiene una estructura de árbol invertido de múltiples niveles que permite un fácil mantenimiento.

- Los archivos de disco y los dispositivos de entrada y salida se tratan como archivos convencionales. Los detalles de los dispositivos se mantienen en el kernel. Eso quiere decir que impresoras, discos, terminales, etc., desde el punto de vista del usuario, se tratan como si fueran archivos normales.

- Desde el principio, los programas fuente estuvieron a disposición del usuario, facilitando en gran medida el descubrimiento y eliminación de deficiencias, así como nuevas posibilidades en su realización.

- Amplia documentación, a manera de mini-manuales.

- Alto desempeño en Servidores.

- Alta configuración en la seguridad del sistema.

- Soporta TCP/IP.

1.3.2 NACIMIENTO DE LINUX

Linux fue desarrollado inicialmente por Linux Torvalds en la universidad de Helsinki (Finlandia) en 1991. El núcleo Linux está basado en MINIX, un pequeño sistema Unix desarrollado por Andrew Tanenbaum con fines académicos. Al principio Torvalds empezó a escribir Linux por pura diversión, sin embargo, a medida que avanzaba en su desarrollo, Linus fue dejando el código fuente de las sucesivas versiones del kernel y utilidades de Linux a disponibilidad de los usuarios de Internet. Este fue sin duda un gran acierto, ya que hizo posible que una multitud de desarrolladores de todo el mundo se familiarizaran con el código, lo cual en primera instancia significó un gran aporte de sugerencias, evolucionado luego hacia un espectacular ejemplo de desarrollo distribuido de software: centenares de desarrolladores independientes, desde diferentes puntos del planeta tomaron a su cargo la producción de software para Linux, ya sea escribiéndolo desde cero o portándolo desde otras plataformas Unix. Esta modalidad de desarrollo continúa aún hoy y

ha permitido a Linux alcanzar un alto nivel de desarrollo y madurez, así también como un amplio grado de aceptación.

El escrito con el que Torvalds convocó a programadores a unirse al proyecto Linux se reproduce a continuación:

"¿Añoras aquellos tiempos con Minix-1.1 cuando los hombres eran hombres y escribían sus propios drivers de dispositivo? ¿No tienes ningún proyecto y deseas hincarle el diente a un sistema operativo para adaptarlo a tus necesidades? ¿Te frustras cuando todo funciona bajo Minix? ¿No quieres perder más noches poniendo en marcha un apestoso programa? Entonces puede que este mensaje sea para ti."

"Como ya comenté hace un mes, estoy desarrollando una versión de libre distribución de un sistema similar a Minix para ordenadores 386-AT. Al fin he alcanzado un estado en el que el sistema incluso puede ser utilizado (dependiendo de lo que desees), y dejaré todos programas fuente de libre distribución. Es solamente la versión 0.02 … pero he conseguido ejecutar con éxito bash, gcc, gnu-make, gnu-sed, compress, etc. bajo él. "

1.3.3 DISTRIBUCIONES DE LINUX

Las primeras versiones de GNU/Linux consistían básicamente de un núcleo y algunas herramientas del proyecto GNU. Con la ayuda de la comunidad de Internet se fueron incorporando paquetes para hacer más flexible el sistema operativo. Con el tiempo, estudiantes, programadores, individuos y compañías, empezaron a distribuir Linux con su propia selección de paquetes. Fue así como nació el concepto de "distribución". Una distribución de Linux está formada por el núcleo Linux con un conjunto de paquetes seleccionados, con herramientas específicas de configuración, empaquetamiento, documentación, compilación, etc. Las herramientas que suelen incluirse en las distribuciones de Linux se obtienen de diversas fuentes, incluyendo de manera importante proyectos de código abierto o libre, como el GNU o el KDE. La mayoría de los sistemas "Linux" incluyen también herramientas procedentes de BSD y de muchos otros proyectos como Mozilla, Perl, Ruby, Python, PostgreSQL, MySQL, Xorg, casi todas con licencia GPL o compatibles.
Gracias a los paquetes suministrados por las distribuciones de Linux es fácil utilizar este sistema operativo para casi cualquier actividad en el ámbito informático, por ejemplo existen utilidades para desarrollo, base de datos, servidores, edición de audio y video, etc.

Múltiples lenguajes de programación están disponibles bajo Linux. Sin duda el principal de ellos es GNU C/C++, pero también es posible

desarrollar en Java, Objective-C, Pascal, LISP, BASIC, Perl, Ada, Eiffel, FORTRAN, Forth, Prolog, Oberon, Simula, Modula-2 y Modula-3, Smalltalk, y algunos otros. Hay varios IDEs disponibles para Linux incluyendo, Anjuta, KDevelop, Ultimate++, Code::Blocks, NetBeans IDE y Eclipse. Además existen editores como lo es Emacs que hoy en día sigue siendo ampliamente utilizado. GNU/Linux también dispone de capacidades para lenguajes de guión (script), aparte de los clásicos lenguajes de programación de shell, la mayoría de las distribuciones tienen instalado Python, Perl, PHP y Ruby. Igualmente, existen varios motores de bases de datos que pueden utilizarse bajo Linux, por ejemplo, MySQL, Postgresql, Oracle, Sybase, FireBird, etc.
Linux ofrece una variada gama de posibilidades a la hora de interconectarse con otros servidores. Como es usual en plataformas Unix, Linux cuenta con soporte nativo de TCP/IP, incluyendo la capacidad para actuar como cliente o servidor NFS (Network File System). El kernel incluye soporte para IPX, lo que le permite funcionar como ruteador en redes Novell. También es posible montar en el sistema de archivos de una máquina Linux volúmenes de un servidor Novell y acceder a sus colas de impresión.

Por medio de la utilización del paquete Samba, Linux puede interactuar con servidores Windows y estaciones Windows. Esto incluye la capacidad para acceder desde Linux a recursos compartidos desde maquinas Windows (directorios e impresoras), como también la posibilidad de exportar directorios e impresoras desde Linux y accederlas desde Windows.

1.3.4 SOFTWARE LIBRE

Linux esta catalogado como software libre y abierto ya que cualquiera que lo desee puede revisar su código fuente y hacerle modificaciones, así como también distribuirlo sin pagar regalías. La promesa de este tipo de software es construir aplicaciones de mejor calidad, de alta confiabilidad, flexibilidad y bajo costo.

El Software Libre le permite al usuario el ejercicio de cuatro libertades básicas:

1. Ejecutarlo con cualquier propósito
2. Estudiar como funciona y adaptarlo a sus necesidades
3. Distribuir copias
4. Mejorarlo, y liberar esas mejoras al público

El software libre tiene la restricción del copyleft, lo que significa que, cualquiera que redistribuya el software, con o sin cambios, debe dar las mismas libertades que antes, y con el requisito de permitir el acceso al código fuente

Existe un movimiento mundial llamado OpenSource que tiene como finalidad el mejoramiento del software a través de la distribución y modificación libre de las aplicaciones. Este movimiento tiene diez premisas para que un programa pueda considerarse OpenSource.

Las características que un software debe cumplir para ser OpenSource se enlistan a continuación:

1. Libre redistribución. El software debe poder ser regalado o vendido libremente.
2. Código fuente. El código fuente debe estar incluido u obtenerse libremente.
3. Trabajos derivados. La redistribución de modificaciones debe estar permitida.
4. Integridad del código fuente del autor. Las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.
5. Sin discriminación de personas o grupos. Nadie puede dejarse fuera.
6. Sin discriminación de áreas de iniciativa. Los usuarios comerciales no pueden ser excluidos.
7. Distribución de la licencia. Deben aplicarse los mismos derechos a todo el que reciba el programa.
8. La licencia no debe ser específica de un producto. El programa no puede licenciarse solo como parte de una distribución mayor.
9. La licencia no debe restringir otro software. La licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.
10. La licencia debe ser tecnológicamente neutral.

CAPÍTULO 2

PROTOCOLOS DE CORREO ELECTRÓNICO

2.1 PROTOCOLOS EN LAS COMUNICACIONES

Cuando se escucha la palabra protocolo, inmediatamente lo asociamos con las reglas que se establecen en ceremonias de carácter diplomático y reuniones de alta sociedad. Sin embargo, en la vida diaria las personas tienen la necesidad de establecer ciertas normas para que se puedan relacionar y comunicar con sus semejantes. Por ejemplo, en la cultura occidental estrechar la mano es señal de saludo, pero en otras culturas esa misma señal tendría un significado totalmente distinto. Algo parecido ocurre en el mundo de las comunicaciones entre equipos electrónicos, debido a la diversidad de fabricantes y máquinas, se deben establecer rigurosas normas de comunicación. A todas esas normas se les conoce como protocolos de comunicaciones.

Para diseñar una red de datos, se debe seleccionar una topología física, esto es, la forma en que se colocan los cables y cómo estos se conectan a los medios electrónicos, una vez hecho lo anterior, se necesita establecer un método de acceso al cable, es decir, se deben establecer los protocolos para acceder al medio y así poder transmitir la información. Al final, esa información debe ser comprendida por todos los elementos que conforman la red, para ello existen reglas en la estructura de la información enviada. Debido a lo anterior, todas las redes de comunicaciones están basadas en protocolos o normas que definen cómo se prepara un mensaje a enviar, cómo se abre la comunicación y cómo se maneja esa comunicación una vez establecida. Esto quiere decir que, un emisor, un receptor y un canal de comunicación no son suficientes para que exista una verdadera comunicación entre computadoras, además de los elementos mencionados, es necesario establecer protocolos entre estos dispositivos para que la información pueda ser recuperada y comprendida.

En la actualidad, la mayoría de los protocolos aceptados y estandarizados se encuentran divididos en capas o niveles. Cada nivel contiene normas y procedimientos que se corresponden a cada etapa en un modelo de comunicaciones.

2.2 ARQUITECTURA DE COMUNICACIONES EN CAPA

En los últimos años, el mundo de las comunicaciones ha tenido una evolución vertiginosa como respuesta a las crecientes necesidades de los usuarios y en algún tiempo las consecuencias del cambio fueron negativas. Al principio, la arquitectura de comunicaciones se diseñaba dependiendo de los dispositivos que se iban a utilizar, con ello se pretendía optimizar el funcionamiento de la red. Estos diseños

presentaban un gran problema ya que al estar ligados fuertemente a los componentes de hardware utilizados no se podían implementar en sistemas con diferentes componentes físicos. Además, cualquier cambio o actualización de un componente físico traía como consecuencia la reprogramación de alguna parte de la arquitectura. Estos modelos de comunicación fueron imposibles de adoptar en redes con máquinas diferentes e incluso, en el caso de máquinas iguales resultaba muy cara la implementación ya que con frecuencia se tenía que reprogramar los protocolos como resultado de algún cambio menor en los dispositivos usados. De esta manera, surgió la necesidad de diseñar un nuevo modelo que independizara las partes de la arquitectura que no dependían de los dispositivos físicos de aquellas que trataban directamente con el hardware. Este tipo de arquitecturas reciben el nombre de arquitecturas estructuradas ya que dividen el modelo en niveles. Para implementar la comunicación, se debe pasar por cada uno de estos niveles. Los niveles están diseñados hasta donde sea posible con independencia del hardware utilizado y sólo el nivel físico implementa características de los dispositivos con los que trabaja.

Las arquitecturas estructuradas de comunicaciones brindaron ciertas ventajas:

- Independencia de Hardware. Gracias al diseño en niveles se evita, en la medida de lo posible, tratar directamente con dispositivos y es sólo el nivel físico el que se diseña en función del hardware. El diseño puede ser adaptado a varios dispositivos físicos mediante un controlador (driver). Un controlador es un programa que permite que una aplicación pueda comunicarse con diferentes tipos de dispositivos, ya que la comunicación se realiza a través de este elemento de software y no de manera directa entre la aplicación y el hardware.
- Facilidad de Actualización Tecnológica.
- Mejora de la Eficiencia.
- Reducción de Costos.
- Interconexión de redes heterogéneas.

Con las ventajas mencionadas, los fabricantes notaron que lo más aconsejable es utilizar las arquitecturas estructuradas, en éstas, las funciones que permiten la comunicación son tratadas por medio de niveles, cada uno de los cuales tiene asociado un conjunto específico de tareas relacionadas por la función desempeñada y por el grado de abstracción de los datos que manejan.

Aunque un modelo de capas está conformado por múltiples niveles, cada nivel sólo puede apoyarse en los servicios brindados por el nivel inmediato inferior el cuál realizará tareas más básicas y más cercanas a los dispositivos físicos (por ejemplo un nivel bajo trata con la transmisión de bits mediante pulsos eléctricos). Podemos decir que un nivel usa los servicios del nivel inmediato inferior y provee servicios al nivel inmediato superior. Gracias a que se tienen las tareas determinadas por nivel, idealmente, los cambios realizados en un nivel, no afectarán los niveles superiores, así se logra que este modelo de comunicaciones sea independiente de los elementos que conforman la red, esto es que sea lo menos dependiente posible de las computadoras y los dispositivos participantes en la comunicación.

En una arquitectura de comunicaciones en capas, las tareas de nivel son ejecutadas por entidades (elementos de hardware o de software), estas entidades necesitan reglas para comunicarse con entidades del mismo nivel pertenecientes a otro sistema. Para ello se utilizan los protocolos de nivel.

En un modelo que sigue la arquitectura estructurada de comunicaciones, el diálogo se realiza entre entidades de un mismo nivel o lo que se le conoce como entidades pares. Sin embargo, las entidades pares no están conectadas directamente (el nivel físico es el único que conecta las entidades pares) es por eso que cada entidad envía sus datos al nivel inmediato inferior y éste a su vez, los envía a su nivel inferior. Obviamente, después de una serie de envíos entre niveles los datos llegan a la entidad par requerida. Por ejemplo, si se quiere enviar un fichero en un modelo de este tipo, la aplicación se comunicará con el nivel superior encargado de la transmisión de ficheros, la entidad de nivel superior tomará el fichero y le añadirá información propia del nivel, a continuación se lo pasará al nivel inferior, este nivel le agregará información propia y lo transmitirá a un nivel más bajo. Este proceso se repite hasta llegar al nivel físico que transmitirá la información a través del medio de conexión. La información que se le añade al fichero depende del nivel, por ejemplo, se agrega información para comprobar que el fichero llega de manera correcta, se añaden cabeceras de control, se checa el orden de los datos, etc. A la unión de datos que se suministran a determinado nivel más la información de control que se añade, se le llama unidad de datos del protocolo de nivel. Además, cuando una entidad toma información de un nivel superior y le añade información de control propias de su nivel, se dice que se lleva a cabo un proceso de encapsulación.

Finalmente, una vez que los datos han pasado por el medio y llegan al equipo deseado, el proceso es inverso, se empieza en la capa más baja y se va quitando información de cada nivel haciendo las respectivas comprobaciones, para al final determinar si la comunicación ha sido exitosa en cada una de las capas.

La arquitectura estructurada de comunicaciones resulta muy conveniente, por lo que se ha creado un modelo de comunicaciones común, el cual se le conoce como modelo de referencia OSI, hoy en día, el modelo OSI es muy popular en el mundo de las redes ya que provee las pautas para el desarrollo de arquitecturas de comunicaciones.

2.2.1 MODELO DE COMUNICACIONES OSI

El modelo OSI (Open System Interconnection) fue desarrollado entre los años 1977 y 1983 por la Organización Internacional de Estándares, este modelo nació por la necesidad de tener un estándar en la arquitectura de comunicaciones y se apega al modelo de comunicaciones estructurado, ya que está conformado por niveles. El modelo OSI fue ideado con la finalidad de tener una base para el desarrollo de sistemas de comunicación, este modelo muestra los servicios que debe ofrecer cada nivel. En otras palabras, indica la función que desempeñará cada nivel y la cooperación que mostrará con niveles adyacentes, pero no dice como se deben implementar esos servicios en el hardware. Ésta característica lo hace independiente del hardware y del tipo de red que se use, por lo tanto, ha sido adoptado como un estándar internacional y es una guía para el diseño de arquitecturas de comunicaciones.

Los fabricantes de productos de red se ajustan a los estándares cuando desarrollan sus productos y el modelo OSI es uno de los estándares mayormente utilizados, es más, se puede decir que es la guía mejor conocida y más utilizada en entornos de red. Además, el modelo OSI ayuda a localizar problemas en redes porque indica el supuesto funcionamiento de las comunicaciones en cada uno de sus niveles.
El conocimiento del modelo OSI es un paso fundamental para la comprensión del funcionamiento de las comunicaciones en red, con lo cual, cualquier individuo interesado en las redes debe empezar haciendo una revisión de este modelo.

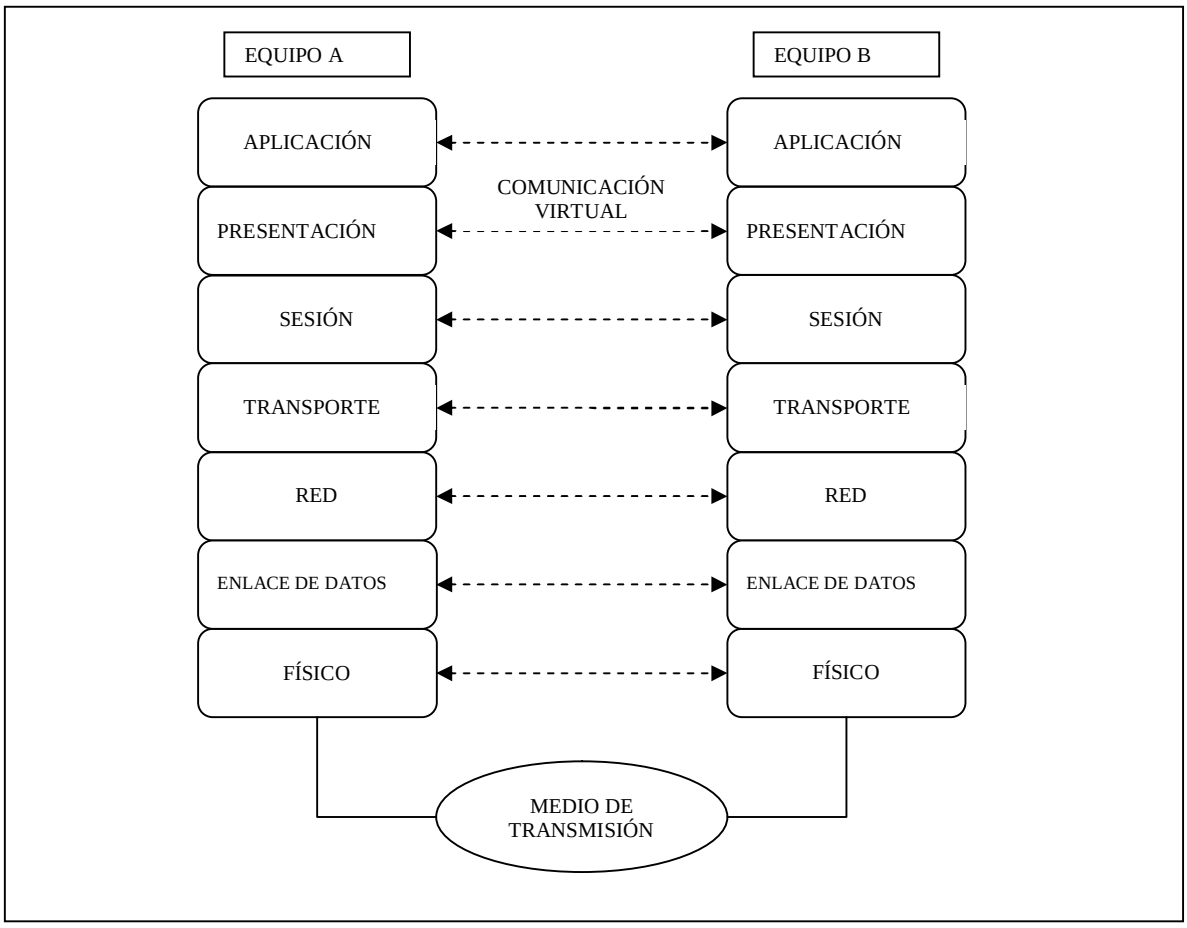


Figura 2.1 Modelo de Comunicaciones OSI.

Como se muestra en la figura, la arquitectura del modelo de referencia OSI se divide en siete niveles. Cada nivel cubre funciones específicas y todos esos niveles muestran cómo se lleva a cabo la comunicación entre diferentes dispositivos.

2.2.1.1 RELACION ENTRE NIVELES DEL MODELO OSI

Cada nivel requiere servicios del nivel inmediato inferior y proporciona servicios al nivel inmediato superior, además, cada nivel parece estar comunicado directamente con su homólogo en el otro equipo, pero en realidad sólo es una comunicación virtual. La comunicación real entre niveles sólo se lleva a cabo en el nivel físico.

Cuando se quiere transmitir información en una red, en el equipo transmisor la información desciende a través de cada una de las capas del modelo. En cada nivel se agrega información de formato de direccionamiento el cual es necesario para la correcta transmisión del paquete a través de la red. En el equipo receptor, el paquete hace su

recorrido en orden inverso, en cada uno de los niveles se va eliminando información relacionada con la función de la capa, así, cuando el paquete alcanza el nivel de aplicación la información referente al direccionamiento ha sido borrada y el paquete se encuentra tal y como fue enviado. Es importante mencionar que los niveles se encuentran separados por fronteras a las cuales se les llama interfaces, toda la interacción entre niveles adyacentes se realiza mediante las interfaces. Una interfaz define los servicios que ofrece el nivel inferior para el superior, además, define como se accede a esos servicios. Cada nivel se basa en los servicios y actividades del nivel inferior.

Se distinguen siete niveles en el modelo OSI:

Nivel de Aplicación

Es también conocido como nivel 7, el más alto en este modelo. Es el acceso a través del cual los agentes de las aplicaciones entran al modelo OSI y una vez en él, pueden solicitarle los servicios de red. A menudo las aplicaciones no se conectan directamente con los servicios ofrecidos por este nivel, sino que lo hacen por medio de programas llamados agentes, que funcionan como intermediarios entre los usuarios y el modelo OSI.

Nivel de Presentación

También conocido como nivel 6, tiene como finalidad resolver las diferencias en el formato y la representación de los datos en un entorno de red.

En redes heterogéneas existen distintos dispositivos, muchos de ellos no trasmiten los datos de la misma manera, es aquí donde entra este nivel, haciendo una traducción y reordenación de bytes. Lo que realmente hace este nivel es tomar los datos y transformarlos a un formato intermedio estándar para que después pueda ser interpretado fácilmente en el nivel de aplicación. Podemos decir que este nivel se encarga de la traducción de datos, la encriptación de los datos, la conversión del conjunto de caracteres y la compresión de datos.

Nivel de Sesión

También llamado nivel 5. Se encarga de la apertura, la utilización y el cierre en la conexión de una sesión (diálogo). En otras palabras este nivel se encarga de la gestión del diálogo entre dos entidades de red,

determina si el dialogo es simultáneo o es alternado (sólo una entidad transmite a la vez). Además este nivel implementa lo que se llama checkpoint, este mecanismo consiste en ir colocando puntos de control en el flujo de datos, estos puntos de control dividen la información en partes más pequeñas, de esta manera, si la red falla, sólo se transmiten las partes que se encuentran después del último punto de control, evitando así, la retransmisión del mensaje completo y esto trae como resultado el ahorro de tiempo.

Nivel de Transporte

La función del nivel 4 (nivel de transporte) es asegurar que el flujo de información sea entregado sin errores, en secuencia y sin pérdidas o duplicados. Lo que hace este nivel en el equipo transmisor es segmentar la información en paquetes para su transmisión sobre la red. En el equipo receptor, este nivel agrupa los paquetes para tener el flujo de información completo y envía una confirmación de que se recibió el mensaje. Si por alguna razón llega un paquete duplicado, el nivel de transporte será capaz de reconocerlo y descartarlo.

Nivel de Red

Este nivel encamina los datos que se quieren transmitir a través de los diferentes nodos hasta llegar a la máquina destino. El nivel de red se encarga de la traducción de las direcciones lógicas en direcciones físicas. Además, es responsable de establecer la ruta que deben tomar los datos desde el emisor hasta el destino. Finalmente, gestiona los problemas de tráfico en la red porque controla la congestión de datos y da prioridad a servicios. Este es el nivel 3 en la jerarquía OSI.

Nivel de Enlace de datos

El principal objetivo de este nivel es ofrecer una comunicación eficiente y fiable, para ello utiliza funciones de control de flujo, detección y corrección de errores. Por lo que proporciona una línea libre de errores al nivel de red. Este nivel controla las variables eléctricas, es decir, la representación eléctrica de los datos.

El nivel de enlace de datos transmite tramas, una trama es una estructura organizada en la que se pueden colocar los datos. Cuando una trama es enviada se espera una confirmación del receptor. En caso de que exista un problema en la transmisión este nivel es capaz de detectarlo. Las tramas dañadas o las que no han recibido confirmación serán enviadas nuevamente.

Nivel Físico

También llamado nivel 1. Es el nivel más bajo en la estructura del modelo OSI, es responsable del medio por el cual los dispositivos se comunican. Este nivel define todos los aspectos relacionados con el hardware en la red, por ejemplo, los conectores, la tarjeta de red, el medio de transmisión, etc. El nivel físico es responsable de la transmisión de señales eléctricas o luminosas a través del medio físico. Por lo tanto, es en este nivel cuando se transmiten (transmisor) o se reciben (receptor) bits puros y éstos bits son el resultado de toda la información que se ha ido añadiendo a través de las distintas capas del modelo OSI. Es importante recordar que este nivel es el único que se encuentra conectado de manera real.

2.2.2 TCP/IP

A principios de los años 80 se adoptó un conjunto de protocolos para la red ARPANET (red que conectaba bases militares, centros de investigación, universidades y laboratorios de gobierno). Ese conjunto de protocolos fueron denominados TCP/IP debido a dos protocolos que forman parte de este grupo, TCP (Protocolo de Control de transmisiones) e IP (Protocolo de Internet), gracias a que estos protocolos fueron implementados en las versiones de Unix que imperaba en ese entonces, TCP/IP se expandió rápidamente. Además, ARPANET derivó en lo que hoy conocemos como Internet y éste utiliza TCP/IP como pila de protocolos estándar. Internet permite la comunicación entre redes heterogéneas, debido a que implementa TCP/IP. De hecho la principal ventaja de TCP/IP es la interoperabilidad entre distintos tipos de equipos, otra ventaja es que, la mayoría de redes permiten trabajar con este conjunto de protocolos. De esta forma, no importa que los equipos no sean del mismo fabricante o que no estén sobre el mismo sistema operativo, si se implementa TCP/IP las computadoras hablarán el mismo lenguaje y por lo tanto serán capaces de comunicarse. Actualmente TCP/IP se ha convertido en la familia de protocolos más extendido, soportado por la mayoría de los fabricantes de los sistemas operativos y con ello es un estándar de facto para la interconexión de redes.
Las ventajas de TCP/IP son:

- Es un estándar. Al ser un estándar es abierto, en otras palabras, no está ligado a una compañía en particular.
- Interconecta Sistemas Operativos. La conexión entre los equipos no dependen del sistema operativo que esté utilizando cada uno.

- Arquitectura Escalable. Los protocolos pueden ampliarse si las necesidades así lo requieren.

2.2.2.1 ESTRUCTURA DE TCP / IP

En una arquitectura de comunicaciones existen varias operaciones independientes: transformar la información en un formato establecido, agrupar la información, determinar el camino que debe seguir esa información, regular la cantidad de información que se transfiere de acuerdo con el medio de transmisión, reunir la información recibida de tal forma que quede ordenada, notificar al emisor de la correcta recepción de los datos, manejo de errores, entre otras. Estas operaciones mencionadas representan un grado de complejidad grande para ser tratadas como una unidad, por esa razón los diseñadores de TCP/IP agruparon esas funciones en niveles independientes pero capaces de sostener algún tipo de comunicación. Cada nivel utiliza los servicios de otros niveles en la misma máquina. Los niveles están organizados de una manera jerárquica y cada nivel solamente puede utilizar los servicios de las capas inferiores en la jerarquía.
La arquitectura TCP/IP está compuesta por cuatro niveles:

- Nivel de Aplicación.
- Nivel de Transporte.
- Nivel de Internet.
- Nivel de Interfaz de Red.

Nivel de Aplicación

Es el nivel más alto en la jerarquía TCP/IP. En este nivel están los protocolos que suministran a las aplicaciones de usuario facilidades de comunicación. Gracias a ello los usuarios pueden utilizar correo electrónico, transferencia de archivos, acceso remoto, etc.

Nivel De Transporte

Este nivel se encarga de establecer y mantener una comunicación entre dos equipos. Proporciona control de flujo, secuencia en los paquetes y notificación de la recepción de dichos paquetes, además, gestiona la retransmisión de paquetes. En función de las necesidades de transmisión, este nivel puede utilizar el protocolo TCP o el UDP (Protocolo de Datagramas de Usuario).

Protocolo de Control de Transmisión (TCP)

A este protocolo se le conoce como protocolo orientado a conexión porque establece una conexión entre dos máquinas antes de transferir información. TCP es el responsable de dividir la información en paquetes y controlar el envío a través de la red, asegurando que llegue de una manera fiable a su destino lo cual implica que la información llegue sin errores, en secuencia y completa.

Protocolo de Datagramas de Usuario (UDP)

A diferencia de TCP, UDP es un protocolo no orientado a conexión con lo cual es mucho menos fiable que TCP porque no garantiza la entrega de datos, sin embargo, puede llegar a ser más rápido que TCP.

Nivel de Internet

Este nivel se encarga de que la información pueda viajar entre diversas redes, en otras palabras, se encarga del encaminamiento. El protocolo más famoso de este nivel es el IP (Protocolo de Internet) que brinda facilidades de direccionamiento y encaminamiento. Al transmitirse un paquete, el protocolo IP le añade una cabecera, así, cada paquete está compuesto por una dirección de origen y destino, un identificador de protocolo, una suma de prueba (checksum) y un tiempo de vida del paquete en la red (TTL). IP es un protocolo no orientado a conexión y envía paquetes (denominados datagramas en este nivel) sin esperar confirmación por parte del receptor, por lo tanto no es fiable.

Nivel de Interfaz de Red

Este nivel proporciona acceso a la red de comunicaciones, controla los dispositivos físicos involucrados en la comunicación y está estrechamente relacionado con el medio de transmisión. Por lo tanto, este nivel es la interfaz con la red física y proporciona un medio para la comunicación entre la arquitectura de red y el nivel Internet.

2.2.3 TCP/IP Y EL MODELO DE REFERENCIA OSI

Existen diferencias entre el modelo de referencia OSI y la familia de protocolos TCP/IP. Hay algunas capas que existen en el modelo OSI y no se contemplan en TCP/IP, por ejemplo, TCP/IP no tiene nivel de presentación, con lo cual las funciones de este nivel deben ser cubiertas por el nivel de aplicación. El nivel de transporte de TCP/IP realiza

funciones del nivel de sesión mediante paquetes. El nivel de Internet es capaz de manejar el nivel de red del modelo OSI mediante una interfaz de red.
Otra diferencia es que en el modelo OSI la comunicación entre los niveles se lleva a cabo con el inmediato inferior y en TCP/IP un nivel se puede apoyar en cualquier nivel inferior.
Por otra parte, el modelo OSI es más que nada conceptual ya que existen muy pocas aplicaciones utilizadas en la actualidad. Sin embargo, OSI es un modelo que se toma como punto de partida en el diseño de arquitectura de comunicaciones, de ahí su importancia. Por el contrario TCP/IP está ampliamente aplicado en la actualidad, prueba de ello es Internet y todo el conjunto de servicios que puede ofrecer gracias a la implementación de muchos protocolos de TCP/IP.

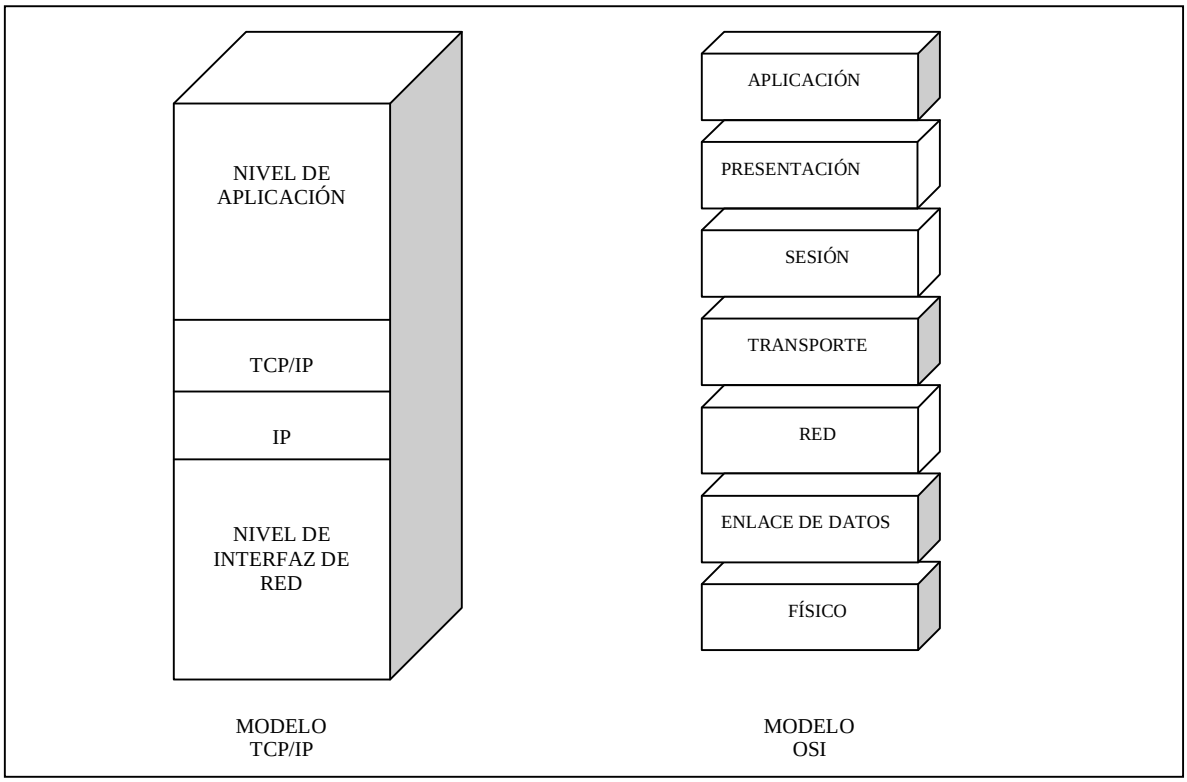


Figura 2.2 Diferencias entre el modelo OSI y el modelo TCP/IP.

2.3 SERVICIOS TCP/IP

La familia de protocolos TCP/IP ofrece múltiples servicios al usuario, incluso, es posible que exista más de una aplicación usando los protocolos TCP/IP a la vez. Para ello es necesario tener un método que identifique los datos asociados a cada aplicación. El mecanismo que usa TCP/IP es lo que se le conoce como puerto, un puerto no es más que una palabra de 16 bits que identifica hacia qué aplicación deben dirigirse los datos, de esta forma, se puede estar utilizando el servicio de correo electrónico y de transferencia de archivos simultáneamente.
TCP/IP se basa en el modelo Cliente/Servidor el cual se divide en dos partes: una es el programa cliente y su contraparte es el programa servidor. El programa cliente es el que hace la petición del servicio y el servidor es la máquina o programa que provee dicho servicio. Ambos programas pueden estar corriendo en la misma máquina aunque lo normal es que se encuentren en diferentes computadoras. Es por ello que cuando se establece una conexión TCP debe existir un punto de comunicación, a este punto se le conoce como socket, un socket está formado por un puerto y una dirección de nodo.

Obviamente para que exista la comunicación entre el programa cliente y el programa servidor se tiene que llevar a cabo una conexión previa y que ambos programas hablen el mismo protocolo.

En TCP/IP existen muchos protocolos de aplicación que han facilitado de manera muy importante la comunicación a nivel mundial. En el presente trabajo sólo se describirán de manera general los protocolos que intervienen en el servicio de correo electrónico.

RFC (Request for Comments)

Los RFC son documentos que contienen investigaciones, innovaciones y metodologías relacionadas con las tecnologías de Internet.

La sociedad de Internet, los ingenieros así como científicos en computación publican documentos acerca de nuevos conceptos, información o convenios en cuanto a tecnologías de la información. The Internet Engineering Task Force (IETF) adopta algunas propuestas de los documentos RFCs publicados y los hace estándares de Internet. Todos los estándares se encuentran en los documentos RFC, pero no todos los documentos RFC son estándares.

2.4 SMTP (SIMPLE MAIL TRANFER PROTOCOL, Protocolo Simple de Transferencia de Correo)

Es el protocolo más usado para el envío de correos a través de Internet, su objetivo principal es transferir correo de una manera confiable y eficiente. SMTP tiene la capacidad de trasferir mensajes de correo a través de Internet, lo que significa que no importa si los elementos involucrados en la transmisión-recepción se encuentran en la misma red o distribuidos en redes diferentes (siempre y cuando soporten el protocolo). El protocolo SMTP se encuentra definido en el RFC 821.

Un correo puede ser entregado de manera directa o pasando a través de varios servidores mediante el protocolo SMTP. Si un cliente SMTP tiene un mensaje para transmitir debe establecer una conexión con el servidor SMTP.

Una vez que el canal de transmisión ha sido establecido se lleva a cabo un saludo entre cliente y servidor, después, el cliente inicia la transmisión de correo valiéndose de una serie de comandos que permiten especificar el origen y destino de correo, así como el cuerpo del mensaje. El servidor proporciona una respuesta por cada comando recibido, el tipo de respuesta indica si el comando se acepta, se esperan comandos adicionales o el comando generó un error temporal o permanente. En caso de que la transmisión del correo resulte exitosa, el cliente puede cerrar la conexión o iniciar una nueva transacción de correo. Con lo dicho anteriormente, se concluye que la tarea del cliente es transferir correos a uno o más servidores SMTP o en su defecto reportar un error de la imposibilidad de la transferencia.

Normalmente la transferencia de correo se lleva a cabo entre la máquina cliente y la máquina servidor, pero también existen casos en que la maquina servidor sólo actúa como puente para los mensajes, en este caso el servidor se convierte en cliente para reenviar el correo (relay). Es por ello que un mensaje de correo puede viajar a través de diferentes maquinas hasta llegar a su destino final.

2.4.1 FASES DEL PROTOCOLO SMTP

Inicio de Sesión

Un servidor SMTP está a la escucha de peticiones por el puerto 25, una sesión SMTP se inicia cuando un cliente contacta al servidor y éste le responde con un código más un mensaje, dicho mensaje puede contener

la identificación y versión del software servidor que implementa el protocolo.

Inicio de comandos por parte del cliente

Después de que el servidor ha enviado el mensaje de bienvenida y el cliente lo ha recibido, el cliente envía el primer comando SMTP el cual normalmente es HELO.

Trasferencia del correo

La transferencia del correo se lleva a cabo en tres pasos. Inicia con el comando MAIL que es enviado por el emisor junto con el campo FROM que indica la dirección del remitente a quien serán enviados los mensajes en caso de algún error en la transferencia del correo. Es importante señalar que este comando le sugiere al servidor que está comenzando una nueva transacción y por lo tanto debe vaciar los buffers de memoria con la finalidad de dejar espacio para el mensaje entrante. La respuesta del servidor al mensaje MAIL será 250 OK si todo va bien o en su defecto regresara un mensaje de error. En caso de que el comando MAIL tenga éxito, el emisor puede enviar uno o varios comando RCPT TO especificando en cada uno de ellos, la dirección del destinatario de correo electrónico. El servidor comprueba que los comandos son correctos, en este caso envía 250 OK, de lo contrario enviara un mensaje de error “550 no such user here".

El tercer paso se da con el envío del comando DATA por parte del cliente, con esta orden el cliente está en condiciones de enviar el cuerpo del mensaje, el servidor responde con “354 Start input mail” e indica la secuencia de caracteres para finalizar el mensaje de correo. A partir de este momento, todo lo que envíe el cliente hasta encontrar la secuencia de caracteres indicada por el servidor será considerado parte del mensaje. Si el servidor recibe el mensaje de manera correcta responderá con un “OK 250” en caso de que exista algún problema notificara con un mensaje de error apropiado.

Finalización de la sesión

Una vez que se ha transmitido el mensaje de manera correcta, el cliente puede iniciar una nueva transferencia o cerrar la comunicación con el servidor. Para hacer esto último se introduce el comando QUIT. Entonces, el servidor mandará una respuesta afirmativa con lo que se da por finalizada la sesión SMTP.

Ejemplo de una sesión SMTP:

```
Trying 10.16.17.123...
Connected to buford.hackenbush.com.
Escape character is '^]'.
220 buford.hackenbush.com ESMTP Postfix
helo woofgang.dogpeople.org
250 buford.hackenbush.org
mail from:<mick@dogpeople.org>
250 Ok
rcpt to:<groucho@hackenbush.com>
250 Ok
data
354 End data with <CR><LF>.<CR><LF>
Subject: Test email from Mick
Testing, testing, 1-2-3...
.
250 Ok: queued as F28B08603
quit
221 Bye
Connection closed by foreign host.
```

2.4.2 COMANDOS Y RESPUESTAS

Los mensajes intercambiados en una sesión SMTP se componen de caracteres ASCII. Los comando enviados por el cliente deben finalizar con la secuencia <CRLF> la cual representa la pulsación de la tecla ENTER (que produce un retorno de carro y un avance de línea). Las respuestas numéricas enviadas por el servidor están formadas por tres dígitos, el primero de ellos indica la categoría de la respuesta.

Las categorías son las siguientes:

- 2XX, la operación solicitada mediante el comando anterior ha sido concluida con éxito
- 3XX, la orden ha sido aceptada, pero el servidor está pendiente de que el cliente le envíe nuevos datos para terminar la operación
- 4XX, para una respuesta de error, pero se espera a que se repita la instrucción.
- 5XX, para indicar una condición de error permanente, por lo que no debe repetirse la orden.

Existen otros comandos en el protocolo SMTP que proporcionan cierta funcionalidad en la transferencia del correo, a continuación se muestran estos comandos así como los códigos de respuesta SMTP.

COMANDOS SMTP

Comando	Descripción
HELO	Identifica el remitente al servidor.
MAIL FROM	Es el inicio para una trasferencia de correo e identifica al emisor.
RCPT TO	Indica el destinatario del correo. Si existen múltiples destinatarios es necesario repetir el comando.
DATA	Permite enviar el mensaje de correo. Cada línea del mensaje tiene un tamaño máximo de 1000 caracteres y debe terminar con la secuencia <CRLF>. Para finalizar el mensaje se inicia con el carácter punto "." seguido de <CRLF>.
RSET	Aborta la transferencia de correo actual.
NOOP	No operación.
QUIT	Pide al servidor que envíe una respuesta positiva y cierra la conexión.
VRFY	Pide al servidor que confirme que un nombre identifica a un destinatario válido.
EXPN	Pide al servidor la confirmación de una lista de correo y que devuelva los nombres de los usuarios de dicha lista.
HELP	Pide al servidor información sobre los comandos disponibles.
TURN	El emisor solicita que se inviertan los papeles. Ahora el cliente actúa como servidor y el servidor como cliente.
SOML	Si el destinatario está conectado, entrega el mensaje directamente a la terminal, en caso contrario lo entrega como correo normal.
SAML	Entrega del mensaje en el buzón

	del destinatario. En caso de que esté conectado también lo entrega en la terminal.
SEND	En caso de que el destinatario esté conectado, entrega el mensaje en la terminal.

2.4.3 RESPUESTAS SMTP

CODIGO	DESCRIPCIÓN
211	Estado del Sistema
214	Mensaje de Ayuda
220	Servicio Preparado
221	Servicio cerrando el canal de transmisión
250	Solicitud completada con éxito.
251	Usuario no local, se enviara a...
354	Inicio del mensaje, finalice con <CRLF>.<CRLF>
421	Servicio no disponible
450	Solicitud de correo no ejecutada porque el buzón está ocupado.
451	No se ejecuta el comando porque existe un error local de procesamiento
452	No se ejecuta el comando, insuficiente espacio de almacenamiento en el sistema.
500	Comando no reconocido.
501	Error de sintaxis en parámetros
502	Comando no implementado.
503	Secuencia de comandos no reconocida.
504	Parámetro no implementado.
550	Solicitud no ejecutada, el buzón no está disponible
551	El usuario no es local, probar...

552	Acción de correo abortada
553	Solicitud no realizada debido a un error de sintaxis.
554	Error en la transferencia.

2.4.4 DIRECCIONES DE CORREO ELECTRÓNICO

Cuando se quiere enviar un correo electrónico se necesita una identificación tanto para el remitente del mensaje como para el receptor del mismo, ésta identificación tiene una función muy parecida a la que tiene en el correo convencional porque ofrece información acerca de la fuente del mensaje y hacia dónde se dirige.

A pesar de que algunas direcciones de correo de Internet pueden parecer muy complejas, todas siguen un patrón bien definido. El formato es el siguiente:

nombre_buzon@dominio1.dominio2.dominioN

La dirección de correo se compone de dos cadenas divididas por el símbolo “@” (arroba).

La parte izquierda de la dirección es lo que se conoce como buzón de correo, el buzón está relacionado con un usuario, un grupo de usuarios o un servicio. Normalmente la cadena que se corresponde con el buzón es el identificador del usuario con que se accede al sistema (login). La parte de la dirección que se encuentra a la derecha se le llama dominio y puede contener varias cadenas separadas entre sí por el símbolo “.” cada una de estas cadenas identifican a un subdominio o al nombre de la máquina. La cadena situada más a la derecha es el dominio de más alto nivel. Por ejemplo la siguiente dirección:

pepe@cancun.fi-a.unam.mx

se interpreta de la siguiente forma
- pepe → nombre del buzón
- cancún → nombre de la máquina
- fi-a → subdominio de menor jerarquía
- unam → subdominio
- mx → dominio de mayor jerarquía (en este caso indica el país dónde se puede localizar la computadora que gestiona el correo).

En algunas ocasiones la parte derecha de la dirección no indica el nombre de la computadora que maneja el correo, en lugar de ello sólo contiene nombres de dominios y subdominios. También es posible que solamente aparezca el nombre del buzón, en este caso el correo se entregará en la misma máquina en la que fue enviado, es por ello que no necesita más información que el nombre de un buzón.

2.5 POP

El protocolo *POP (Post Office Protocol, Protocolo de oficina de correo)* se utiliza para recuperar correo electrónico desde un servidor hasta una máquina cliente. A menudo, los mensajes son almacenados en la máquina cliente y borrados del servidor, aunque este comportamiento puede cambiarse.

Cuando un cliente desea recuperar mensajes a través del protocolo POP3 establece una conexión TCP con el servidor. Una vez que la conexión ha sido realizada, las máquinas involucradas inician un diálogo con la finalidad de llevar a cabo la transacción de forma exitosa. El cliente y el servidor intercambian una serie de comandos hasta que la conexión finaliza de una manera adecuada o cuando la conexión es abortada por alguno de ellos.

Los comandos definidos en el protocolo POP3 consisten en una palabra reservada seguida de uno o varios argumentos. Para finalizar un comando, se utiliza la secuencia de caracteres <CRLF> que es un salto de línea y retorno de carro (tecla ENTER). El diálogo completo sólo utiliza elementos ASCII imprimibles con lo que se pueden ver tanto las peticiones del cliente así como las respuestas del servidor.

Muchos usuarios de correo electrónico acceden a sus mensajes mediante POP ya que no necesitan tener una conexión permanente a la red, POP3 permite descargar los mensajes a una máquina local desde el servidor para su posterior revisión. Lo anterior, ayuda a los usuarios que cuentan con conexiones lentas o intermitentes.

2.5.1 FASES EN EL PROTOCOLO POP3

Normalmente el servidor que proporciona el servicio POP3 escucha peticiones a través del puerto 110.

Existen básicamente tres estados en una sesión POP3:

El cliente inicia una conexión al puerto 110, el servidor envía un saludo de bienvenida como respuesta. En este punto, la sesión entra en lo que se conoce como estado de autorización (AUTHORIZATION STATE).

Durante este estado el cliente debe identificarse ante el servidor enviando el nombre de usuario así como la contraseña. El servidor comprueba la información recibida y aceptará o rechazará la conexión dependiendo si los datos son correctos. Una vez que los datos han sido validados satisfactoriamente, se llega al estado de transacción (TRANSACTION STATE) donde se prepara toda la información relativa al cliente, para que éste pueda manipular el buzón de correo a través de comandos válidos, es precisamente en este estado cuando el cliente puede ver, descargar y marcar mensajes para su eliminación. Es importante mencionar que los mensajes serán eliminados hasta que se teclee el comando de salida (QUIT).

Después de las operaciones hechas por el cliente y cuando se teclea el comando QUIT, la sesión entra en el estado de actualización (UPDATE STATE) en el cual se cierra la sesión de manera ordenada y por lo tanto se cierra también la conexión TCP.

2.5.2 SESIÓN POP3

Cuando una sesión POP3 entra al estado de autorización el cliente debe identificarse enviando su nombre de usuario como primer dato y a continuación su contraseña. El nombre de usuario se precede del comando USER y la contraseña va precedida del comando PASS. La respuesta del servidor a cada uno de estos comandos depende de la validez de los datos.

Después de la exitosa identificación de un usuario, se llega a lo que se conoce como estado de sesión. Para recuperar los mensajes en una sesión POP3, se utiliza el comando RETR acompañado del número del mensaje que se pretende leer. De forma similar se pueden borrar los mensajes. El comando que se utiliza para borrar los mensajes es DELE seguido del número de mensaje a borrar.

Una vez que el cliente ha recuperado sus mensajes y quiere abandonar la sesión necesita ejecutar el comando QUIT para entrar a la fase de actualización. Es aquí donde el correo del usuario se desbloquea y se eliminan todos aquellos mensajes que fueron marcados para borrar.

Obviamente, el usuario casi nunca ve los comandos y respuestas de una sesión POP3 porque el agente que utiliza para gestionar el correo esconde todos los detalles de la transacción. Si un usuario quiere ver los detalles de una sesión POP3 puede utilizar la herramienta telnet para lograr su objetivo.

A continuación se muestra una sesión POP3.

```
$ telnet correo 110
Trying 192.168.1.3...
Connected to correo.servidor.com.
Escape character is '^]'.
+OK POP3 correo.servidor.com v7.64 server ready
USER prueba
+OK User name accepted, password please
PASS password
+OK Mailbox open, 1 messages
RETR 1
+OK 531 octets
>From prueba  Wed Aug  8 14:38:46 2001
Return-Path: <luis@uni.edu>
Delivered-To: prueba@correo.servidor.com
Received: from speaker.servidor.com (speaker.servidor.com [192.168.1.1])
        by correo.servidor.com (Postfix) with SMTP id EB2A01A2BD
        for <prueba@correo.servidor.com>; Wed,  8 Aug 2001
14:38:26 -0400 (EDT)
Message-Id: <20010808183826.EB2A01A2BD@correo.servidor.com>
Date: Wed,  8 Aug 2001 14:38:26 -0400 (EDT)
From: luis@uni.edu
To: undisclosed-recipients:;
Status:

This is a test message.

.
DELE 1
+OK Message deleted
QUIT
+OK Sayonara
Connection closed by foreign host.
```

La sesión anterior muestra claramente que si se usa el comando USER y PASS, la información enviada queda expuesta a cualquier intrusión ya que los datos viajan tal y como se han escrito, quedando al descubierto tanto el login como el password del usuario. Para evitar lo anterior, existe un comando opcional, llamado APOP, que permite identificar a un usuario evitando los problemas de seguridad de USER y PASS.

El estado de transacción es donde el cliente puede recuperar los mensajes, la siguiente tabla muestra los comandos que se pueden utilizar durante esta fase.

Comando	Descripción
STAT	Solicita el estado del buzón. El

	servidor responde afirmativamente y devuelve el número de mensajes que hay en el buzón así como el tamaño que ocupan (bytes).
LIST	Se utiliza para solicitar las estadísticas del buzón. Se puede invocar pasando como argumento el número de mensaje del cual se desea saber la estadística. El servidor devuelve una respuesta afirmativa con el número de cada mensaje y el tamaño de cada uno. En el caso de indicar un número de mensaje erróneo, el servidor devuelve una respuesta negativa.
RETR	Permite recuperar un mensaje del buzón siempre que éste no esté marcado para borrar. El comando se invoca pasando como argumento un número de mensaje. La respuesta del servidor será afirmativa si el número del mensaje es válido. El servidor enviará el mensaje solicitado. Si el número del mensaje es inválido, el servidor enviará una respuesta negativa.
DELE	Marca un mensaje para eliminar. La eliminación se produce cuando se entra en la fase de actualización.
NOOP	El servidor no hace nada ante este comando, sólo responde afirmativamente. Este comando permite mantener activa la conexión POP.
RSET	Elimina las marcas de los mensajes señalizados para borrar mediante el comando DELE.

TOP	Permite recuperar cierto número de líneas de un mensaje que se le indica como argumento.
UIDL	Solicita una cadena que identifique cada uno de los mensajes almacenados.

Tabla 2.1 Comandos POP3.

2.6 PROTOCOLO IMAP

El Protocolo de Acceso a Mensajes (IMAP, Internet Message Access Protocol) permite a las aplicaciones clientes acceder y manipular los mensajes de correo electrónico sobre un servidor remoto. IMAP autoriza a los clientes a manipular carpetas de mensajes remotos como si fueran locales, a estas carpetas se les denomina "mailbox". El protocolo IMAP proporciona medios para crear, renombrar y eliminar "mailboxes". El protocolo IMAP se define en el RFC 2060.

Cuando se quiere acceder a un mensaje en IMAP, se utiliza un número que identifica unívocamente a cada mensaje en la carpeta.
Podemos destacar las siguientes ventajas en el uso del protocolo IMAP:

IMAP lo utilizan principalmente los usuarios que tiene una gran movilidad, es decir, que pueden obtener acceso a su correo desde varias máquinas. IMAP almacena los mensajes en una ubicación central. Así, cualquier cliente IMAP que esté autorizado para realizar una conexión con el servidor remoto podrá leer sus mensajes sin importar la ubicación geográfica.

El protocolo IMAP4 permite accesos simultáneos a múltiples clientes y proporciona ciertos mecanismos a los clientes para que se detecten los cambios hechos a un mailbox por otro cliente concurrentemente conectado. A través de la utilización de banderas definidas en el protocolo IMAP4 de los clientes, se puede vigilar el estado del mensaje, por ejemplo, si el mensaje ha sido o no leído, respondido o eliminado. Estas banderas se almacenan en el servidor, de manera que varios clientes conectados al mismo correo pueden detectar los cambios hechos por otros clientes.

Los clientes de IMAP4 pueden crear, renombrar o eliminar correo del servidor, y mover mensajes entre cuentas de correo. El soporte para

múltiples buzones de correo también le permite al servidor proporcionar acceso a los directorios públicos y compartidos.

El protocolo IMAP proporciona un gran control sobre los mensajes, por ejemplo, tiene la habilidad de examinar las cabeceras antes de recuperar el cuerpo del mensaje, de esta manera un usuario decide si es importante el mensaje o prefiere no leerlo. De la misma manera, el usuario puede eliminar el correo electrónico que no le interesa sin tener que ver antes el cuerpo del mensaje, lo cual evita el tener que descargar un mensaje si éste no es importante.

IMAP4 proporciona un mecanismo para que los clientes pidan al servidor que busque mensajes de acuerdo a una variedad de criterios. Este mecanismo evita que los clientes descarguen todos los mensajes de su buzón de correo. Esta forma de trabajar de IMAP puede dar tiempos de respuesta más rápidos para usuarios que tienen una gran cantidad de mensajes o mensajes grandes.

2.6.1 SESIÓN IMAP

Al ser IMAP un protocolo perteneciente a TCP/IP se basa en el modelo cliente-servidor, el cliente se conecta al servidor, y este le responde con un saludo, a partir de este momento existe una interacción entre el cliente y el servidor hasta que la conexión se cierra. La interacción consiste de comandos por parte del cliente y respuestas por parte del servidor ante la petición del cliente. La transmisión de datos entre el cliente y el servidor se lleva a cabo en forma de cadenas finalizadas por la secuencia <CRLF>. Cuando un cliente desea enviar un comando al servidor genera una cadena alfanumérica llamada tag (por ejemplo, 0001, 002, son tags validos), los tags se generan por el cliente en cada comando enviado.

Junto con POP3, el protocolo IMAP es uno de los más usados para recuperar mensajes de correo electrónico, aunque se debe decir que IMAP es más complejo que POP, ya que acepta más comandos y como consecuencia el diálogo entre cliente y servidor es más extenso.
A continuación se muestra una sesión IMAP-4, en esta versión del protocolo un servidor IMAP escucha peticiones por el puerto 143.

```
$ telnet servidor 143
Trying 192.168.1.3...
Connected to servidor.correo.com.
Escape character is '^]'.
* OK servidor.correo.com IMAP4rev1 v12.264.phall server ready
A1 LOGIN prueba password
```

```
A1 OK LOGIN completed
A2 SELECT Inbox
* 1 EXISTS
* NO Trying to get mailbox lock from process 29559
* 1 RECENT
* OK [UIDVALIDITY 997295985] UID validity status
* OK [UIDNEXT 4] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\* \Answered \Flagged \Deleted \Draft \Seen)]
Permanent flags
* OK [UNSEEN 1] first unseen message in /var/spool/mail/prueba
A2 OK [READ-WRITE] SELECT completed
A3 FETCH 1 BODY[HEADER]
* 1 FETCH (BODY[HEADER] {494}
>From prueba  Wed Aug  8 16:02:47 2001
Return-Path: <luis@uni.edu>
Delivered-To: prueba@servidor.correo.com
Received: from speaker.correo.com (speaker.correo.com [192.168.1.1])
        by servidor.correo.com (Postfix) with SMTP id 2C7121A2BD
        for <prueba@servidor.correo.com>; Wed,  8 Aug 2001
16:02:25 -0400 (EDT)
Message-Id: <20010808200225.2C7121A2BD@servidor.correo.com>
Date: Wed,  8 Aug 2001 16:02:25 -0400 (EDT)
From: luis@uni.edu
To: undisclosed-recipients:;

)
* 1 FETCH (FLAGS (\Recent \Seen))
A3 OK FETCH completed
A4 FETCH 1 BODY[TEXT]
* 1 FETCH (BODY[TEXT] {25}
This is a test message.
)

A4 OK FETCH completed
A5 COPY 1 demos
A5 OK COPY completed
A6 LOGOUT
* BYE servidor.correo.com IMAP4rev1 server terminating connection
A6 OK LOGOUT completed
Connection closed by foreign host.
```

2.7 PROTOCOLO MIME (*Multipurpose Internet Mail Extensions*, Extensiones de Correo Internet Multipropósito)

El protocolo estándar para la transmisión de mensajes de correo electrónico en Internet es el protocolo SMTP, pero debido a que SMTP fue diseñado hace mucho tiempo, no contempló las necesidades de comunicación de hoy en día. El protocolo SMTP intercambia mensajes en un formato que no permite elementos multimedia, como lo son: imágenes, sonido y video, incluso, no es posible escribir mensajes en algunos idiomas. La solución a la limitante de codificación de SMTP es el protocolo MIME.

MIME es un protocolo estándar de Internet, su función es determinar el formato que se le debe dar a los mensajes para que puedan ser

intercambiados entre diferentes sistemas de correo. MIME permite casi cualquier tipo de formato en los mensajes, por ejemplo, es posible transmitir texto, audio, video, aplicaciones, etc.

El protocolo MIME proporciona mecanismos para que se pueda enviar diversos tipos de información por correo, por ejemplo, proporciona codificación de caracteres y contenido binario de ocho bits. De esta manera, permite ampliar la capacidad de representación de mensajes. Algunas de las características de MIME son las siguientes:

- Permite el envío de múltiples cuerpos dentro del contenido.
- Permite añadir contenido de cualquier tipo, no sólo texto.
- Permite utilizar un juego de caracteres diferentes del US-ASCII.

Para que MIME lleve a cabo su función, es necesario añadir una serie de campos a la cabecera de los mensajes SMTP. Estos campos son:

- Versión MIME.
- Tipo de Contenido.
- Codificación utilizada.
- Identificación y descripción del contenido.

Cada mensaje MIME incluye datos que informan al receptor sobre el tipo de datos y la codificación utilizada.

Cuando un mensaje de correo transporta información utilizando MIME, la cabecera del protocolo SMTP se ve modificada con la información aportada por MIME. Esta información tiene el aspecto siguiente:

From: jose@ejemplo.mx
To: juan@dominio.mx
MIME-Versión: 1.0
Content-Type: image/gif
Content-Transfer-Encoding: base64

Los dos primeros campos son propios del protocolo SMTP, que definen al emisor y al receptor del mensaje, el tercer campo es la versión del protocolo MIME, el cuarto campo indica el tipo de datos que se está enviando, en el ejemplo se trata de una imagen gif. El último campo indica qué tipo de codificación se ha utilizado para convertir la imagen en una representación ASCII de 7 bits, por ejemplo, base64.

Tipos de contenido en MIME (Content-Type)

Content-Type	Descripción
Text	Texto. Permite especificar el conjunto de caracteres utilizado
image	Para datos de imágenes estáticas
audio	Para grabaciones de sonido
video	Para grabaciones de video
application	Para envío de programas ejecutables
message	Para mensajes de correo completos o referencias externas
multipart	Para envío de mensajes múltiples

El tipo multipart permite cuatro subtipos:

Tipo	Descripción
Mixed	Indica que un mensaje tiene partes independientes, con tipos y codificación diferentes
Alternative	Indica que el mensaje contiene distintos formatos de representación de la misma información
Parallel	Indica que el mesaje incluye subpartes que deben ser ejecutadas al mismo tiempo
digest	Indica que un mensaje contiene un conjunto de mensajes

A pesar de que el protocolo HTTP no interviene directamente con el sistema de correo, la mayoría de los servidores actuales implementan el servicio de webmail. Este servicio debe estar montado en un servidor web que implementa el protocolo http.

2.8 PROTOCOLO HTTP

En la actualidad, el servicio más extendido de Internet junto con el correo electrónico es el WWW. El World Wide Web, es un servicio que

reúne dos potentes técnicas: por un lado la búsqueda de información y por el otro el hipertexto. Tras la idea de la creación del hipertexto surgió también la idea de la hipermedia, en la cual, mediante los enlaces no sólo es posible acceder a otros documentos, sino también a imágenes, animaciones, video, sonido, etc. Para poder navegar por la WWW, se creó el protocolo HTTP (HiperText Transfer Protocol).

HTTP es un protocolo de nivel de aplicación utilizado para el intercambio de información hipermedia dentro de la WWW. HTTP funciona generalmente sobre TCP/IP y las conexiones se realizan (normalmente) al puerto TCP 80.

HTTP es un protocolo que se basa en la filosofía cliente/servidor. Un cliente envía un mensaje, compuesto por un comando, un identificador de recurso y la versión del protocolo, seguido del mensaje, el cual contiene generalmente información sobre el cliente. La respuesta que envía el servidor está formada por una línea de estado que incluye la versión del protocolo y un código de respuesta que determina de qué modo se llevó a cabo la operación. Después de lo anterior se coloca el cuerpo del mensaje.

La conexión entre el cliente y el servidor debe ser iniciada por el cliente y cerrada por el servidor tras el envío de la respuesta a la solicitud, sin embargo, tanto los clientes como los servidores deben tener en cuenta que una conexión puede cerrarse de manera prematura debido a una acción del usuario, por el vencimiento de los temporizadores o simplemente debido a un fallo en el programa.

2.8.1 VERSIÓN DEL PROTOCOLO

HTTP utiliza un sistema de numeración del tipo <*major*>.<*minor*> para indicar las versiones del protocolo. Mediante este mecanismo, el emisor puede indicar el formato del mensaje y su capacidad para entender futuras comunicaciones HTTP. El número <*minor*> se incrementa cuando los cambios realizados al protocolo añaden características que no cambian el formato general del algoritmo utilizado para analizar sintácticamente el mensaje, pero que sí dotan de mejora adicionales al emisor. El número <*major*> se incrementa cuando el formato del mensaje dentro del protocolo ha cambiado.

2.8.2 CODIFICACIÓN DE LA TRANSFERENCIA

Este concepto indica qué tipo de codificación se le aplica al mensaje para asegurar su transferencia de manera fiable. Se utiliza una codificación de tipo “chunk”, en la que se modifica el cuerpo del mensaje para que éste sea transferido en bloques (chunks), cada uno de los cuales incluye un indicador de tamaño.

2.8.3 TIPOS DE MENSAJES

Los mensajes pueden ser de dos tipos:

Mensajes que van desde el cliente hasta el servidor. (Solicitudes)
Mensajes que van desde el servidor hasta el cliente. (Respuestas)

Los mensajes están formados por una cabecera (que contiene una serie de campos) y un cuerpo donde se incluye la información en sí. El cuerpo va separado de la cabecera por medio de una línea en blanco.

Cabeceras de los Mensajes

Las cabeceras de los mensajes pueden ser de varios tipos:

- Cabeceras generales
- Cabeceras de petición
- Cabeceras de respuestas
- Cabeceras de entidades

Cada campo dentro de la cabecera tiene el formato siguiente:

Identificador: valor <CRLF>

El orden en que se envían los campos de cabecera carece de importancia, sin embargo, se suelen enviar los campos de las cabeceras generales en primer lugar, seguidos de los campos de las cabeceras de petición, a continuación los campos de las cabeceras de respuesta y por último los campos de las cabeceras de identidad.

CABECERAS GENERALES

Identificador: Valor	Descripción
Date: fecha	Utiliza el formato estándar Mon,

	02 Nov 1999 n10:30:00 GMT
MIME-Version: versión	Versión MIME de las cabeceras: MIME-Version 1.0
Pragma: directiva	Especifica una directiva concreta de la implementación. Pragma: no caché. Indica a un proxy que obtenga una nueva versión del elemento aunque ya exista en la caché y no se guarda en caché

CABECERAS DE PETICIÓN

Identificador: Valor	Descripción
Accept: tipos	Permite describir qué tipos de datos se aceptan como respuesta
Accept-Charset: tipo	Permite describir qué tipos de caracteres de aceptan en la respuesta
Accept-Encoding:tipo	Permite restringir los valores de codificación
Accept-Language:leng	Permite definir en qué lenguaje se desea obtener la respuesta
Authorization:cred	Incluye la información de identificación (credenciales) del cliente para acceder a recursos protegidos
From: id	Identifica el origen del mensaje
Host:host:puerto	Identifica el host y el puerto donde se encuentra el recurso
If-modified-Since: fecha	Se utiliza para que el comando GET sea condicional. Si el elemento no se ha modificado devuelve un código 304 sin cuerpo
If-Match:condición	Permite establecer una condición
If-None-Match:condición	Permite establecer una condición negada
If-Range: condición	Permite recuperar parte de una entidad si cumple la condición

If-Unmodified-Since: fecha	Se utiliza para que el servidor ejecute la acción si el recurso no ha sido modificado desde la fecha indicada
Max-Forwards: número	Se utiliza con el método TRACE para limitar el número máximo de proxys que pueden reenviar el mensaje
Refer:URL	La identidad del elemento de donde se obtuvo el enlace
User-Agent. Producto	Identifica al software del cliente

CABECERAS DE RESPUESTA

Identificador:Valor	**Descripción**
Age: tiempo	Tiempo estimado desde que la respuesta fue generada
Location: URL	Localización preferida del servidor para este elemento
Proxy-Authenticate	Este campo tiene que ser incluido cuando se envía un código de error 407
Public:lista_métodos	Lista los métodos soportados por el servidor
Retry-after: tiempo	Indica el tiempo estimado que el servicio estará no operativo
Server: Producto	Identifica el software del servidor
Vary: tipo	Indica que la entidad de respuesta fue seleccionada entre las diferentes respuestas disponibles
Warning: información	Permite incluir información adicional sobre la respuesta
WWW-Authenticate: acuerdo	Datos que identifican el sistema de autenticación y acuerda con el cliente que se autentifique el mismo

CABECERA DE ENTIDAD

Identificador:Valor	Descripción
Allow: método	Lista de métodos que admiten un recurso
Content-encoding: tipo	Si se utiliza codificación indica el algoritmo utilizado
Content-Language: leng	Describe el lenguaje natural de la información
Content-Location: URI	Identifica el lugar del recurso que viaja en el mensaje
Content-length: tamaño	Tamaño del cuerpo a transferir
Content-MD5: md5	Proporciona detección de la alteración del mensaje
Content-Range: bytes	Se inserta en los mensaje cuando éstos se cortan para indicar qué posición ocupan en el mensaje original
Content-type: tipo	Tipos que establece la IANA
Etag: etiqueta	Define la etiqueta de la entidad
Expires: fecha	Fecha tras la cual el elemento dejará de tener validez
Last-modified: fecha	Ultima vez que se modificó el elemento

MENSAJES

Solicitudes

Los mensajes que van del cliente al servidor reciben el nombre de solicitudes. Las solicitudes pueden ser simples o completas.

Una solicitud simple consta de un método y un identificador Uniforme de Recursos (URI, cadena de texto que identifica unívocamente un recurso en la red). Una solicitud completa consta de una línea de solicitud seguida de una serie de cabeceras, a continuación una línea en blanco y posteriormente el cuerpo del mensaje.

Métodos

Cuando un cliente hace una petición debe especificar un método.
Los métodos son los siguientes:

OPTIONS: Permite al cliente consultar qué métodos hay asociados a un recurso.
GET: Descarga un elemento.
HEAD: Solicita ver las cabeceras que se enviarán con un método concreto.
POST: Envía un elemento al servidor.
PUT: Solicita colocar la información enviada en la URI identificada.
DELETE: Solicita al servidor que elimine la entidad indicada por la URI.
TRACE: Permite obtener una réplica del mensaje enviado, de esta forma el cliente sabe con certeza qué recibió el servidor.

Respuestas

Los mensajes que viajan desde el servidor hacia el cliente se denominan respuestas. Una respuesta puede ser simple o completa. Una respuesta simple sólo debería ser enviada como respuesta a una solicitud simple. Una respuesta completa consta de una línea de estado, seguido por la cabecera general, cabecera de respuesta, a continuación la cabecera de entidad y finalmente el cuerpo del mensaje separado por una línea en blanco. La primera línea de una respuesta completa es la línea de estado, la cual consta de la versión del protocolo seguida de un código de estado y su frase asociada.

Códigos de Estado

Código	Descripción
1xx	Informativo. No utilizado, se reserva para usos futuros
2xx	Éxito. La acción fue recibida y aceptada
3xx	Redirección. Se necesita una acción adicional para llevar a cabo la solicitud
4xx	Error del cliente. La solicitud contiene sintaxis errónea o no se puede conceder
5xx	Error del servidor. Error del

	servidor ante una solicitud aparentemente correcta

Los valores individuales de los tipos de código son los siguientes:

Código	**Mensaje Asociado**
200	OK
201	Created. Creado
202	Accepted. Aceptado
204	No Content. No Contenido
205	Reset Content. Reiniciar Contenido
206	Partial Content. Contenido Parcial
300	Multiple Choices. Opciones múltiples
301	Moved permanently
302	Moved temporarily
304	No modificado
305	Utilizar proxy
400	Solicitud errónea
401	No autorizado
402	Se requiere pago
403	Prohibido
404	No encontrado
405	Método no Permitido
406	No aceptable
407	Se requiere autenticación de proxy
409	Conflicto
411	Se requiere longitud
413	Entidad solicitada es demasiado grande
414	URI solicitada es demasiado grande
500	Error interno del servidor
501	No implementado
502	Gateway erróneo
503	Servicio no disponible

CAPÍTULO 3

AGENTES DE CORREO ELECTRÓNICO

3.1 SERVICIO DE CORREO ELECTRÓNICO

El correo electrónico es de los servicios más usados en la red, desde sus inicios provocó una verdadera revolución en la forma en que los usuarios intercambian información por medios electrónicos. Para la mayoría de las personas el intercambio de mensajes a través del correo electrónico es un proceso simple, el emisor escribe un mensaje, lo envía dando una orden y en pocos segundos, como por arte de magia el correo aparece en la bandeja del destinatario. Sin embargo el proceso para la entrega del mensaje es más complejo de lo que parece.

En su forma más básica el servicio de correo electrónico se divide en dos componentes principales: MTA y MUA.

MTA (Mail Transfer Agent, Agente de Transferencia de Correo)

Es el software que actúa como servidor de correo, este componente es capaz de entregar, redireccionar y almacenar correo electrónico, además es el componente que entiende el protocolo SMTP por ello es el corazón del servicio de correo. Algunos ejemplos de MTA son: Sendmail, Exim, Qmail, Postfix, Exchange, Zmailer, etc.

MUA (Mail Agent User, Agente de Correo de Usuario)

Es el programa que actúa como cliente de correo, ya que proporciona una interfaz que permite a los usuarios leer, escribir, enviar y almacenar mensajes de correo. Ejemplos de MUA son: Eudora, Outlook, Pine, Thunderbird,etc.

Si bien es cierto que los dos componentes anteriormente comentados son la base del correo, en algunos sistemas podemos encontrar un tercer componente llamado Agente de Entrega de correo (MDA, Mail Delivery Agent), su principal función es entregar el correo de forma local, una vez que el MTA ha enrutado los mensajes le pasa el control al MDA para que sea éste quien deposite los mensajes en los buzones de los usuarios, además puede proporcionar funciones de filtrado.

Cuando un usuario trata de enviar un mensaje de correo electrónico, se llevan a cabo ciertos pasos de los cuales el usuario casi nunca es consciente ya que suceden de forma interna, los pasos más comunes en el envío-recepción de un mensaje de correo son los siguientes:

1. El emisor redacta el mensaje con un agente de correo de usuario,

especifica el destinatario en el campo To, el asunto del mensaje en el campo Subject y luego escribe el texto del mensaje, una vez completado los campos le ordena al MUA que transmita el mensaje.

2. Para que el mensaje se pueda transmitir es necesario que se convierta en un formato específico, es por ello que se le añaden ciertos campos como son un identificador de mensaje, la fecha y otros elementos importantes para la entrega. En este punto el MUA inyecta el mensaje en el MTA.

3. Si la inyección tiene éxito el mensaje es responsabilidad del agente de transporte de correo. Si el destinatario se encuentra en la misma máquina donde opera el MTA, el correo se pasa al agente de entrega de correo para que sea éste quien deposite el mensaje en el buzón del destinatario. Caso contrario, si el destinatario no es local, lo primero que se hace es una consulta al DNS para resolver el nombre del servidor donde se encuentra el destinatario, después se manda el mensaje a ese servidor a través de comandos SMTP. Puede suceder que el servidor que indica la dirección de correo sólo sea un servidor de puente y que el buzón final se encuentre en otro equipo, si este es el caso, entonces el proceso de envío del mensaje se repetirá hasta que se alcance la maquina destino y el mensaje sea depositado en el buzón del usuario final, o en su defecto sea rechazado. Un mensaje puede ser rechazado por diversos motivos, el destinatario no existe, el destinatario no tiene espacio suficiente en su buzón, el mensaje no contiene campos adecuados, etc. Sin embargo, si el mensaje alcanza el buzón final, permanecerá en esa localización hasta que sea leído por un MUA.

Todos los MTA tienen como objetivo transportar correo electrónico ya sea de forma local o hacia otras máquinas, sin embargo, no siempre es posible la entrega inmediata del mensaje, cuando ocurre este evento el servidor de correo almacena el mensaje y lo trata de entregar cada cierto intervalo. Al conjunto de mensajes que se almacenan para su posterior entrega se le llama cola de correos y se ubican en un archivo dedicado para ese fin. Todos los MTA's proporcionan comandos para gestionar la cola de correos ya que para todos ellos la cola es un elemento vital en el intercambio de correos. Existe otro elemento común para la mayoría de los Agentes de Transporte de Correos, este elemento es conocido como alias. Un alias es un sobrenombre para una cuenta de correo, existen muchos casos en que es necesario mandar

correos a un usuario que no tiene cuenta como tal en el sistema de correos, en este caso se usan alias para redirigir el correo. Por ejemplo cuentas especiales o demonios del sistema no se corresponden a ningún individuo, por lo tanto este tipo de cuentas se tienen que hacer coincidir con una cuenta de usuario mediante un alias. No sólo se puede redirigir el correo a una persona, se puede relacionar un nombre lógico con una serie de cuentas del sistema de correo, es así como se puede empezar a estructurar una lista de correos, con lo que se deduce que los alias son importantes en cualquier servidor de correo.

En la actualidad la mayoría de los MTA's tienen características muy similares, la diferencia fundamental entre ellos radica en cómo implementan esas características, además de la seguridad, la rapidez, la integración con otras herramientas, el soporte, la facilidad de la implementación, uso, etc.

Existen muchos servidores de correo, a continuación se hará un análisis de los MTA's más usados para sistemas tipo UNIX con licencias que no restringen su uso de forma gratuita: Sendmail, qmail, Postfix y Exim.

3.2 SENDMAIL

Sendmail es uno de los Agentes de Transferencia de Correo más populares de la red, según estadísticas, sendmail se utiliza en la mayoría de los servidores de correo tipo Unix. Fue escrito por Eric Allman a principios de los años ochentas.

El programa "sendmail" es capaz de realizar diversas tareas, todas ellas importantes para el correcto funcionamiento del correo. Puede esperar por la entrada de mensajes, transportar mensajes a otras máquinas, proporcionar los mensajes al agente dedicado de entregar correo localmente, además concatena mensajes en archivos y tiene la capacidad de hacer que un mensaje sea procesado por un programa antes de depositarlo en el buzón.

3.2.1 ARQUITECTURA DE SENDMAIL

A Sendmail se le conoce como servidor de correo monolítico, lo que significa que en un archivo se especifican la mayoría de las configuraciones para el procesamiento del correo electrónico, en otras palabras, las funcionalidades que proporciona sendmail en su mayoría se especifican en un solo fichero.

Sendmail puede trabajar en dos modos: puede actuar sólo cuando se necesite procesar algún mensaje, con lo cual aceptará enviar o recibir mensajes por un determinado tiempo o puede operar en segundo plano (como demonio del sistema), en este último caso, sendmail se ejecuta de modo persistente escuchando por el puerto estándar de correo y esperando mensajes entrantes o salientes.

El sistema sendmail implementa el protocolo ESMTP para el intercambio de correo y aunque es capaz de implementar otros protocolos, ESMTP se ha convertido en un estándar de facto para el procesamiento de correo, ya que su versión más antigua SMTP da muchas limitaciones en la entrega de correo. Además ESMTP incrementa las opciones de seguridad y funcionalidad en el procesado de los mensajes.

3.2.2 CONFIGURACIÓN DE SENDMAIL

Sendmail tiene un archivo de configuración principal llamado “sendmail-cf” Este archivo ha sido criticado por su complejidad, la sintaxis que maneja no es intuitiva, es más, puede llegar a ser muy confusa. Sin embargo, la sintaxis críptica de este archivo es una de las mayores ventajas de sendmail, ya que el archivo está diseñado para que sendmail sea rápido a costa de la incomprensión de los usuarios.

Afortunadamente en las versiones recientes de sendmail, el panorama ha cambiado, ahora es muy común crear el archivo de configuración a partir de directivas de macros m4, apoyándose precisamente de un programa llamado m4 y de un archivo de macros llamado sendmail.mc.

Pasos en la configuración de sendmail.

1. Habilitar las características deseadas en el archivo sendmail.mc.
2. Ejecutar el programa m4 para obtener el archivo sendmail-cf a partir de sendmail.mc.
3. Configurar reglas de entrega de correo editando el archivo mailertable.
4. Configurar reglas de reenvío de correo editando el archivo *access*.
5. Configurar las reglas necesarias para el manejo de múltiples dominios, a través del archivo *virtusers*.
6. Definir sinónimos para los usuarios (alias) a través del archivo *aliases*.
7. Convertir *mailertable*, *access*, *virtusers*, and *aliases* a archivos de datos.

8. Definir todos los nombres de hosts válidos para nuestro equipo editando el archivo *local-host-names*.
9. Reiniciar sendmail.

Características del archivo sendmail.mc

El primer paso en la configuración de sendmail se resume en habilitar ciertas características en el archivo sendmail.mc, este archivo es el auxiliar para crear el archivo de configuración final, para fortuna de los usuarios este archivo tiene un formato muy entendible. A continuación se hace un análisis del formato del archivo de macros y se explican algunas características útiles para la creación del archivo final.

Como ejemplo se tiene un extracto del archivo sendmail.mc

```
divert(0)
#bloque
#de
#cometarios
divert(-1)
dnl This is a comment line
include(`/usr/lib/sendmail-cf/m4/cf.m4')
VERSIONID(`Mail server')dnl
OSTYPE(`linux')
define(`confDEF_USER_ID',``8:12'')dnl
define(`confPRIVACY_FLAGS', `authwarnings,needmailhelo,noexpn,novrfy')dnl
define(`confSMTP_LOGIN_MSG', ` Sendmail')dnl
define(`confSAFE_FILE_ENV', `/var/mailjail')dnl
define(`confUNSAFE_GROUP_WRITES')dnl
undefine(`UUCP_RELAY')dnl
undefine(`BITNET_RELAY')dnl
FEATURE(`access_db',`hash -o /etc/mail/access.db')dnl
FEATURE(`smrsh',`/usr/sbin/smrsh')dnl
FEATURE(`dnsbl')dnl
FEATURE(`blacklist_recipients')dnl
FEATURE(`mailertable',`hash -o /etc/mail/mailertable.db')dnl
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable.db')dnl
FEATURE(`use_cw_file')dnl
FEATURE(`masquerade_entire_domain')dnl
FEATURE(`masquerade_envelope')dnl
FEATURE(`nouucp')dnl
MASQUERADE_AS(`hackenbush.com')dnl
MASQUERADE_DOMAIN(`.hackenbush.com')dnl
EXPOSED_USER(`root')dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
Cwlocalhost.localdomain
```

El primer elemento a resaltar dentro del archivo sendmail.mc es el comentario. Los cometarios empiezan con la cadena *dnl* (delete through newline). La cadena *dnl* puede aparecer al principio de una línea, lo que significa que la línea será ignorada cuando se construya el archivo

sendmail.cf. Además la cadena *dnl* puede aparecer al final de la línea lo que evita líneas en blanco cuando se construye el archivo sendmail-cf. Existe una forma de poner bloques de comentarios, esto se hace con la sentencia *divert*. La directiva *divert*(-1) inicia una sección de comentarios que terminan con *divert(0*), todo lo que este entre estos dos elementos se considera comentario.

La macro VERSIONID brinda información acerca de la versión y el formato se establece a gusto personal. Aquí se puede establecer la información en que fue modificado el archivo y por quién.

La macro OSTYPE carga la configuración requerida para un sistema operativo en especifico, debido a lo cual es muy importante que el valor de esta directiva este correcto.

Otra directiva importante en el archivo sendmail.mc, es la definición de variables m4, las cuales siempre empiezan con la cadena *define* o *undefined*, seguida de un nombre de variable y el valor para esa variables, si es que aplica el caso. Los nombres de variables así como los valores de dichas variables se encierran entre los elementos `', para evitar ser expandidos durante la transformación del archivo. En caso de que las variables sean booleanas, el valor por default para una directiva *define* será verdadero, mientras que una directiva *undefine* a la que no se le especifica un valor, tomará por default el valor falso. Algunos parámetros modificables por la directiva *define* son los siguientes:

confFORWARD_PATH	El valor de esta variable le indica a sendmail el lugar donde se encuentran los archivos .forward
confMAX_HEADERS_LENGTH	Limita el tamaño de la cabecera de los mensajes que permitirá sendmail, el valor se escribe en bytes.
confDOMAIN_NAME	Se utiliza para configurar un nombre de dominio predeterminado, En caso de que no esté disponible el DNS.
conf_LOG_LEVEL	Especifica el nivel de log, para sendmail el mínimo es 0 y el máximo es 13.
conf_MAILER_NAME	El alias usado para los mensajes devueltos. Comúnmente se configura a MAILER-DAEMON, el cual es un alias de

	root.
conf_MAX_MESSAGE_SIZE	El tamaño máximo en bytes de cualquier mensaje aceptado para su envío. Si se configura de un modo adecuado, este parámetro puede prevenir un ataque de denegación de servicio.

La directiva *FEATURE* se encarga de enlistar las diferentes características soportadas por sendmail. Para que una característica sea habilitada de manera correcta, la directiva debe empezar con la cadena *FEATURE* seguida de uno o más parámetros encerrados entre paréntesis y en caso de que exista un valor para el parámetro, se debe encerrar entre los caracteres ``'. Se requiere una instancia *FEATURE* por cada característica a activar. Algunas características que se pueden habilitar a través de la directiva *FEATURE* so las siguientes:

acces_db	Activa una base de datos de remitentes y dominios de mantenimiento para aquellos correos que se rechacen o se devuelvan.
always_add_domain	Añade el dominio dentro de los correos electrónicos que se envíen a través de Sendmail, incluso aquellos que se envíen a usuarios locales
blacklist_recipients	Define una lista de destinatarios a los que no se les permite recibir correos electrónicos.
domaintable	Permite sustituir nombres de dominio.
mailertable	Permite asociar un nombre de servidor de correo diferente con cada dominio virtual permitido.
promiscuos_relay	Autoriza utilizar el servidor como relay para cualquier sitio.
redirect	Redirecciona los mensajes destinados a usuarios que no existan ya en el sistema. Requiere una entrada correspondiente en el archivo de alias.
relay_entire_domain	Permite a todas las máquinas del dominio local enrutar su correo a través del servidor local.
smrsh	Es una shell limitada para otorgar cierta

	seguridad a sendmail cuando se procesan ciertos comandos para determinados mensajes.
use_cw_file	Contiene una lista de todos los alias de DNS para el servidor de correo.
virtusertable	Permite diferenciar cuentas con el mismo nombre pero con diferentes dominios virtuales en el sistema local.

```
MASQUERADE_AS(`hackenbush.com')dnl
MASQUERADE_DOMAIN(`.hackenbush.com')dnl
EXPOSED_USER(`root')dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
Cwlocalhost.localdomain
```

La directiva *MASQUERADE_AS* permite enmascarar el dominio, por ejemplo si un mensaje tiene un nombre de host que coincide con la directiva *MASQUERADE_DOMAIN* ese nombre de host será escrito con el valor de host escrito en la directiva *MASQUERADE_AS.* En algunas ocasiones se quiere evitar que se enmascare el dominio para ciertos usuarios, esto se puede hacer con la directiva *EXPOSED_USER.*

MAILER permite especificar los diferentes agentes encargados del correo electrónico, entre los más importantes se pueden mencionar:

local Se refiere al agente de entrega de correos local, es el agente que entrega correos a usuarios del sistema local.

prog El agente prog lo usa sendmail para enrutar los mensajes hacia otros procesos corriendo en el sistema local.

esmtp Es una extensión del protocolo estándar de correo, además de manejar SMTP, incluye cuerpos de mensaje complejos y contenidos tipo MIME. Por default sendmail maneja este agente.

relay Esta opción se usa para redireccionar los mensajes con el protocolo SMTP a través de otros MTA's.

smtp Al configurar esta opción sendmail sólo utilizara el protocolo SMTP, con la limitante de que los mensajes estarán formados con caracteres que se encuentren en ASCII y que contenga 7 bits.

smtp8 El Agente smtp8 se diseñó para trabajar con sistemas remotos que puedan manejar 8 bits en los mensajes de correo pero que no comprenden completamente el protocolo ESMTP.

Aparte de las opciones comentadas anteriormente, existen muchas otras como pueden ser los dominios virtuales, las listas negras para prevenir el spam, los mecanismos de autenticación, etc. Sin embargo, todas las características tienen un formato similar entre ellas, consisten en directivas y valores para esas directivas.

Generar el archivo sendmail-cf a partir de sendmail.mc

Una vez configuradas las características deseadas en el archivo sendmail.mc, es posible construir el archivo sedmail.cf con un programa llamado m4 el cual ayuda a procesar todas las directivas del archivo de macros m4. Suponiendo que el archivo de macros se encuentra en el directorio /etc/mail, el comando para realizar la construcción del archivo final es:

m4 /etc/mail/**sendmail.mc** > /etc/mail/sendmail.cf

Configurar reglas de reenvío de correo editando el archivo *access*

/etc/mail/access

La información que se encuentra en la base de datos *acces* define qué máquinas o direcciones IP pueden acceder al servidor de correo y qué clase de acceso tienen permitido. Las máquinas se listan junto con las opciones OK, REJECT, RELAY o simplemente junto con un mensaje de error que se entrega a la rutina de gestión de excepciones de sendmail. Los equipos que se listan junto con la opción OK, que es el valor por defecto, tienen permiso para enviar correo al servidor siempre y cuando la dirección de correo de destino sea la máquina servidora de correo. Las máquinas listadas junto con la opción REJECT tienen el acceso prohibido a conexiones de correo electrónico con el servidor. Por último

los equipos que poseen la etiqueta RELAY tienen permitido enviar correo para cualquier destino a través del servidor de correo.

```
enviodespam.com              550 No se acepta Spam
conocidoporspam@             550 No se acepta spam
envio.de.spam                REJECT
maquina.enviodespam.com      OK
132.248.54                   RELAY
```

En el ejemplo se pueden observar cinco entradas. Los generadores de correo que coinciden con la parte izquierda de la tabla se ven afectados por la parte acción especificada en la parte derecha. Los primeros dos ejemplos emiten un código de error para la rutina de excepciones de sendmail. El mensaje de error se transmite a la máquina remota cuando se recibe un correo que coincide con la parte izquierda de la tabla. La siguiente entrada rechaza correo de una determinada máquina de Internet, envio.de.spam. La siguiente entrada acepta conexiones de correo de la máquina maquina.enviodespam.com, lo cual proporciona más información que enviodespam.com. Las coincidencias más completas tienen precedencia sobre las menos específicas. La última entrada permite actuar como “relay” o pasarela de correo electrónico para aquellas máquinas que posean una dirección IP que comience por 132.248.54. Estas máquinas podrían enviar correo destinado a otros servidores de correo a través del servidor configurado con las reglas anteriores. Cada vez que se actualiza este archivo se debe ejecutar el comando *make* dentro del directorio donde se encuentre instalado el árbol de archivos de sendmail para que se actualice la base de datos.

Configurar las reglas necesarias para el manejo de múltiples dominios, a través del archivo *virtusertable*

El archivo virtusertable asocia direcciones de correo pertenecientes a dominios y buzones virtuales con buzones reales. Los buzones pueden ser locales, remotos, alias definidos en el archivo aliases o incluso se pueden asociar con otros archivos.

```
root@hola.com              root
postmaster@hola.com        postmaster@nt.hola.net
@hola.com                  admin
```

En el ejemplo superior se observa una correspondencia para el dominio hola.com. Este fichero se procesa de arriba a abajo buscando la primera coincidencia. La primera entrada asocia root@hola.com con el buzón de correo local denominada root. La siguiente entrada asocia postmaster@hola.com con la carpeta postmaster situada en la máquina

nt.hola.net. Al final, si no se ha encontrado ninguna coincidencia para hola.com se le asigna la última asociación, la cual hace coincidir cualquier mensaje de correo que contenga hola.com con la carpeta de correo local denominada admin.

Definir sinónimos para los usuarios (alias), a través del archivo *aliases*

La base de datos de alias contiene una lista de directorios virtuales que son traducidas a otros usuarios, ficheros, programas o incluso otros alias. A continuación se muestran unos ejemplos de la sintaxis que se puede utilizar dentro del archivo /etc/mail/aliases:

```
root: usuariolocal
postmaster: jose,ivan,luis
no.importa:  /dev/null
nuevo: "|/usr/local/bin/programa"
```

El formato del archivo es simple, el nombre del buzón de correo que aparece a la izquierda de los dos puntos se traduce al destino de la derecha. El primer ejemplo simplemente traduce la carpeta root a la carpeta usuariolocal, la cual se examina de nuevo utilizando la misma base de datos de alias, y si no existe ninguna otra coincidencia el mensaje se entrega al usuario local usuariolocal. En el ejemplo siguiente se muestra una lista de correo. Todo correo que se envía a postmaster se traduce en un envío para tres usuarios locales diferentes: jose, ivan y luis. Es importante señalar que también se pueden especificar carpetas remotas mediante la forma usuario@hola.com. El siguiente ejemplo muestra la escritura del correo a un archivo, en este caso en /dev/null, lo que tiene como consecuencia que el correo se pierda. El último ejemplo muestra el envío de un mensaje de correo a través de un programa, lo que se logra utilizando una barra vertical (pipe).

Convertir *access*, *virtusertable*, and *aliases* a archivos de datos

Los archivos acces, aliases y virtusertable, son archivos de texto para que se puedan manipular por el usuario. Sendmail necesita archivos de datos para que los pueda procesar con facilidad, debido a esto, todos los archivos anteriores se deben convertir en un formato entendible por el agente de correo para ello se utiliza el comando makemap, la sintaxis general es:

```
makemap btree archivo_de_datos < archivo_de_texto
```

Definir todos los nombres de hosts válidos para nuestro equipo editando el archivo *local-host-names*

Este archivo es una lista de nombres de máquinas que sendmail considera como nombres locales. A menudo se utiliza para escribir todos los nombres de los equipos que el servidor de correo considera como host local. Cuando se actualiza este archivo, sendmail necesita ser reiniciado para que surtan efecto los cambios realizados

Iniciar sendmail

Para reiniciar sendmail se puede localizar el identificador de proceso para el demonio y pasarle la señal de reinicio con el comando kill. En algunos sistemas se puede usar el script de inicio.

3.2.3 ARCHIVOS DE SENDMAIL

Cuando se instala sendmail, se crean ciertos archivos útiles en el funcionamiento del correo electrónico, entre los más importantes se pueden mencionar:

/usr/sbin/sendmail	Es un archivo binario de sendmail, se utiliza para controlar los modos de operación del Demonio de correo, este archivo debe pertenecer a smmsp
/etc/mail/sendmail.cf	Archivo principal de configuración de sendmail, este archivo contiene muchas opciones que describen el comportamiento del sistema de correo.
/usr/sbin/newaliases	Es un archivo binario que reconstruye la tabla de los alias del sistema. En realidad este archivo es una liga al archivo binario sendmail al cual se le pasan ciertos parámetros.
/var/spool/mqueue	Directorio que almacena los mensajes para su posterior entrega (mensajes encolados). Este directorio debe pertenecer a root con permisos 700. Este directorio se define en el archivo

	sendmail.cf
/etc/mail/statistics	Archivo que guarda un resumen de las estadísticas del sistema de correo.
/usr/bin/mailq	Comando que permite ver los mensajes que están en la cola de correos.
/var/run/sendmail.pid	Es un archivo que guarda el número de proceso del demonio sendmail, se utiliza cuando se quiere enviar una señal al servicio de correo.

3.2.4 COMANDOS SENDMAIL

Sendmail se maneja a través de su archivo binario el cual lleva el mismo nombre. Existen algunos parámetros útiles que permiten gestionar los diferentes elementos que componen a este gran agente de correos.

sendmail -bp	Comando que permite ver los correos que han sido puestos en la cola para su posterior entrega. El mismo resultado se consigue con el comando *mailq*
sendmail -q	Comando que permite procesar los correos que están en la cola.
sendmail -bi	Regenera el archivo de datos de los alias. Es igual al comando *newaliases.*
sendmail -Carchivo	Especifica un archivo de configuración diferente al que viene por default
sendmail -bt	Comando que hace que sendmail se ejecute en modo de prueba de direcciones, este comando es muy útil para comprobar interactivamente las reglas del archivo de configuración y localizar posibles problemas.

Existen múltiples configuraciones para sendmail, en realidad es un sistema de correos muy versátil que permite tener un sistema de correos altamente personalizado y es por ello que sigue siendo uno de los agentes más usados en la actualidad a pesar que ha sido criticado por su diseño, complejidad y sus problemas de seguridad.
Es importante mencionar que sendmail tiene tantas opciones de configuración que es muy complicado explicar todas sus características así como su funcionamiento. Podría dedicarse todo un libro a las configuraciones y opciones que proporciona sendmail.

A continuación se muestran algunas ventajas y desventajas del servidor de correo sendmail.

Ventajas de Sendmail

- Software maduro.
- Tiene una comunidad de Usuarios grande, por lo que existe gran soporte gratuito.
- Soporte Comercial.
- Amplia documentación.
- Altamente configurable.
- Soporta muchos sistemas operativos, así como múltiples arquitecturas.
-

Desventajas de Sendmail

- Un sistema complejo.
- Se le considera un software con problemas de seguridad.
- Es un sistema monolítico (para muchos usuarios esto es una ventaja).
- Es grande en cuanto a código se refiere.

3.3 EXIM

Exim es un Agente de Transporte de correo creado en la universidad de Cambridge, Inglaterra. Funciona sobre sistemas tipo Unix, incluyendo Linux y aunque es posible instalarlo en sistemas Windows se recomienda no hacerlo si se pretende tener un sistema de correo en producción.

El programa Exim fue escrito principalmente por Philip Hazel, este programador deseaba construir un nuevo MTA, para ello tomó las principales funcionalidades del agente "Smail" y agregó otras características. Debido a que Hazel no tenía la certeza de cómo funcionaría su nuevo sistema de correo lo llamó **EX**perimental **I**nternet **M**ail (Exim).

En la actualidad Exim se ha convertido en uno de los agentes de Transporte de Correo más poderosos, muestra de ello es que la distribución de Linux, Debian, lo ha adoptado como su sistema de correo por default.

Afortunadamente Exim es software libre y se distribuye bajo la licencia GNU-GPL. Exim se encuentra en la versión 4.69 y se prevé la pronto aparición de lo que será la versión 5. La principal fuente para obtener información acerca de Exim se puede encontrar en la página electrónica oficial: www.exim.org.

3.3.1 MODO DE OPERACIÓN DE EXIM

Cuando exim recibe un email, lo primero que hace es aplicar las reglas de reescritura de cabeceras después se comprueba si el destinatario es local o está en otra máquina. Si es local, se pasa por la lista de elementos llamados "directors", hasta que alguno sepa qué hacer con él y lo reparta. Si no es local, se pasa por la lista de elementos llamados "routers", hasta que alguno sepa qué hacer con él. Para saber si un e-mail es local, se compara el dominio del destinatario con la lista definida en el campo ``local_domains'' en el fichero de configuración.

Todas las operaciones en Exim son llevadas a cabo por un único archivo binario, el cual opera de diferentes modos dependiendo de los parámetros que recibe.

Normalmente Exim funciona como un proceso en segundo plano para cumplir dos propósitos:

1. Esperar por conexiones TCP/IP sobre el puerto SMTP estándar,

una vez recibida una petición el programa creará un nuevo proceso para atender la petición. Se puede configurar el número máximo de conexiones que Exim manejará simultáneamente, si ese límite se alcanza, entonces las conexiones nuevas entrantes serán rechazadas.

2. Procesar los correos que están en la cola a determinados intervalos. Cada vez que se intenta entregar un mensaje almacenado en la cola se crea un nuevo proceso, el programa encargado de procesar la cola esperará a que el mensaje sea entregado o en su defecto falle antes de intentar procesar otro mensaje.

3.3.2 CONFIGURACIÓN DE EXIM

Las opciones de configuración proporcionada por el administrador se usan en dos etapas, la primera durante la instalación y la otra cuando Exim actúa como demonio para empezar a procesar mensajes. Esta última configuración es la que controla el comportamiento del Agente de transporte de correo y la mayoría de las características de funcionamiento se encuentran en un archivo llamado exim.conf, razón por la que a Exim se le considera un sistema de correo monolítico.

Cuando se activa Exim, lo primero que hace el demonio es leer su archivo de configuración para determinar las opciones con las que procesará mensajes, dicho archivo gobierna todo el comportamiento del sistema de correo, es por ello que es tan importante su correcta configuración. Una vez activado el servicio de correo, los datos del archivo de configuración se mantienen en memoria mientras el programa de Exim se ejecuta. Si se hace un cambio en el archivo de configuración de Exim ya sea para habilitar o inhibir cierta característica es necesario indicarle al demonio de Exim que vuelva a leer su archivo de configuración. Para indicarle a Exim que lea nuevamente su archivo se debe detener e iniciar el servicio, la otra opción es mandar una señal de refresco al proceso

Al igual que muchos MTAs, Exim adopta la interfaz de comandos de Sendmail e implementa casi todas sus opciones lo que permite familiarizarse rápidamente con Exim si se ha sido un administrador de sendmail.

El Archivo de configuración de Exim

El archivo de configuración de Exim determina la forma en que se procesan mensajes. Este proceso implica la búsqueda de información acerca del destino de los mensajes y cómo transportarlos hasta su destino.

Exim es capaz de manejar la entrega de correo de diferentes maneras. Puede encontrar el destinatario a través de una búsqueda en el DNS, o lo puede hacer buscando una correspondencia en el archivo de alias. Además puede entregar un mensaje usando el protocolo SMTP o concatenando el correo al buzón del usuario correspondiente. Para manejar los distintos comportamientos de Exim, el archivo de configuración se estructura en siete bloques separados lógicamente. Los bloques son los siguientes:

- **Configuración principal**: En esta sección se encuentran las directivas principales de configuración: nombre del Servidor, dominio, máquinas con relay permitido, etc.

- **Transports:** Son los componentes que se encargan de la entrega de mensajes, ya sea concatenando el mensaje a un buzón o enviándolo a través del protocolo SMTP.

- **Directors**: Se encargan de procesar las direcciones de los mensajes para que la entrega sea correcta. “Directors” se encargan específicamente de las direcciones locales.

- **Routers**: Tienen una función muy similar a los Directors, en realidad la única diferencia es que los Routers se encargan de las direcciones remotas mientras que los Directors gestionan las direcciones locales. La diferencia entre Directors y Routers es tan pequeña que posiblemente en un futuro ya no se haga distinción entre ellos.

- **Retry**: En esta sección se especifica el tiempo que tiene que transcurrir hasta que se considere que un mensaje no se puede enviar.

- **Reescritura**: Aquí están las reglas generales de reescritura para ciertos elementos en los mensajes.

- **Autenticación**: En esta sección se describen los mecanismos de autenticación que maneja el Servidor Exim.

Cada uno de los bloques del archivo de configuración cuenta con ciertas variables, por lo que si se desea activar cierta característica basta con proporcionar un valor adecuado a esas variables. Si bien es cierto que Exim tiene todas sus opciones en el archivo de configuración, también necesita de ciertos archivos auxiliares para hacer su trabajo en forma adecuada, por ejemplo, necesita un archivo de alias para manejar nombres lógicos.

3.3.3 PRINCIPALES PROCESOS DE EXIM

Exim utiliza cuatro procesos principales para gestionar los mensajes de correo electrónico: El Demonio, el proceso de recepción, el gestor de la cola y el proceso de entrega de mensajes.

El Demonio

El demonio permanece a la escucha de conexiones entrantes e inicia un proceso de recepción por cada conexión establecida. Además el demonio puede iniciar periódicamente el proceso que se encarga de gestionar la cola. Cabe destacar que sólo hay un demonio de Exim.

Las dos principales tareas del demonio son: escuchar por conexiones entrantes y llamar periódicamente al gestor de la cola. Para lograr ese cometido el demonio se inicia con los siguientes parámetros.

```
exim –bd –q15m
```

La opción –d (daemon) le indica al binario de Exim que inicie un proceso como demonio a la escucha de conexiones SMTP entrantes, mientras que la opción –q (queue) le sugiere procesar la cola el tiempo especificado después de dicha opción, en este caso el proceso gestor de la cola será invocado cada quince minutos.

Cuando el demonio de Exim inicia, se escribe el número de su proceso (PID) en un archivo, el número de proceso es útil cuando se quiere terminar el proceso o mandar una señal de reinicio cada que se modifica el archivo de configuración de Exim.

Existen una serie de opciones en el archivo de configuración de Exim que se relacionan con el demonio, algunas de ellas se listan a continuación.

local_interfaces

En muchos sistemas de correo se tiene más de una interfaz de red, esas interfaces se distinguen gracias a su dirección IP. La opción local_interfaces permite definir sólo aquellas interfaces que aceptarán peticiones SMTP. Si esta opción se deja vacía todas las interfaces de red aceptarán conexiones entrantes SMTP. Para especificar varias interfaces, se escribe la dirección IP de cada una de ellas separadas por comas (,).

queue_run_max

Esta variable controla el número máximo de gestores de cola que el demonio de Exim puede ejecutar simultáneamente. Lo cual no significa que los procesos gestores de cola se inician al mismo tiempo.

smtp_accept_max

Limita el número de conexiones SMTP simultáneas que Exim aceptará. Si ese límite se alcanza, los futuros intentos de conexión serán rechazados con un mensaje de error. El valor de cero indica que no habrá límites en el número de conexiones SMTP entrantes. El valor adecuado para esta variable depende del poder de procesamiento del equipo así como de la rapidez de la red. La variable acepta valores numéricos y por default tiene un valor de veinte.

smtp_accept_max_per_host

Esta opción restringe el número de conexiones SMTP simultáneas desde una misma IP. Cuando el límite se alcanza las conexiones futuras desde esa misma IP serán rechazadas. El valor por default para esta variable es cero (sin restricción).

smtp_accept_queue (default = 0)

Si el número de conexiones SMTP simultáneas rebasa este valor, los mensajes serán puestos en la cola y ningún proceso de entrega será iniciado de forma automática. Obviamente este valor debe ser menor que el de la variable smtp_accept_max.

Proceso de Recepción

Acepta mensajes entrantes y los almacena en el área designada por el fichero de configuración de Exim.

Proceso Gestor de la Cola

El proceso que se encarga de gestionar la cola busca los mensajes que están esperando a ser entregados, para ello inicia un proceso de entrega para cada mensaje El gestor de la cola procesa un mensaje a la vez. Es importante entender que el proceso gestor de la cola no entrega ningún mensaje por él mismo, sino que inicia un proceso de entrega para que sea éste quien haga el trabajo.

Existe una opción en el archivo de configuración de Exim, la cual impide que el proceso de entrega sea iniciado automáticamente, con esto, la única manera de iniciar procesos de entrega será a través del gestor de la cola ya sea de manera periódica o de forma manual. La opción referida es queue_only.

Proceso de Entrega

El proceso de entrega intenta realizar la operación de entrega sobre un mensaje. Este proceso es el más complejo de todos y puede ser iniciado por la llegada de un mensaje, por el gestor de la cola o por un comando del administrador del correo.

La mayoría de los procesos de Exim tienen una corta duración. Estos procesos realizan una tarea, como por ejemplo recibir o entregar mensajes y cuando cumplen su cometido el proceso se da por terminado. La única excepción es el proceso que actúa como demonio ya que éste sigue ejecutándose de manera permanente a la escucha de nuevas conexiones.

3.3.4 ARGUMENTOS DE LÍNEA DE COMANDOS DE EXIM

Siempre que se ejecuta el archivo binario de Exim se le pasan opciones o argumentos que especifican la acción que se pretende ejecutar. Existen muchas opciones de línea de comandos para Exim, estas opciones pueden ser divididas en los siguientes grupos:

Control de Entrada

En este apartado se encuentran las opciones que inician procesos para recibir mensajes entrantes.

exim -bd -oX 1225

Inicia el proceso de Exim como demonio y le indica que escuche por el puerto 1225

Datos Adicionales de Mensaje

Opciones que proporcionan información que se incorpora dentro de los mensajes.

exim –f falsacuenta@falsodominio

Esta opción permite sobrescribir la verdadera dirección.
Opciones del Proceso Gestor de la Cola.

Opciones para iniciar el proceso gestor de la cola, además permiten seleccionar cuáles mensajes serán procesados. Algunos ejemplos de estas opciones son:

-q Procesado normal de la cola.
-qf Gestión de los mensajes de la cola con entregas forzadas
-qff Entregas forzadas incluyendo mensajes no válidos en ese momento.
-ql Procesado de mensajes en la cola y que son del dominio local.

Sobreescritura de la Configuración

Aquí se encuentran las opciones alternas al archivo de configuración normal. Por ejemplo para leer un archivo de configuración que no sea el usual se tiene:

exim -C /etc/exim/alterno.conf

El comando para revisar mensajes mientras están en la cola es el siguiente.

exim –bp

Muestra todos los mensajes que se encuentran en la cola.

Control de Mensajes

Opciones para forzar las entregas de mensajes y manipular otras características de los mensajes almacenados en la cola.

exim –M ID Crea un proceso de entrega para el mensaje con identificador ID.
exim –Mrm Cancela la entrega de un mensaje.
exim Mvp ID Muestra el cuerpo del mensaje con identificador ID.

Pruebas

Opciones para probar el manejo de direcciones de filtrado, expansión de cadenas y reintentos.

exim –bV

Comprueba que el binario puede leer su archivo de configuración en forma adecuada.

Depuración

En esta sección se encuentran las opciones para depurar Exim así como su configuración.

exim –d1 Muestra mensajes de depuración en el error estándar.

3.3.5 ARCHIVOS DE REGISTRO EN EXIM

Exim escribe tres diferentes registros de información: registro principal, registros de rechazo y registros de pánico.

Registro Principal

Colecciona información acerca de la llegada de cada mensaje así como de la entrega de cada correo. El formato de los registros son compactos con la intención de mantener el archivo lo más pequeño posible.

Registro de rechazo

En este archivo se guarda información relacionada con los mensajes rechazados debido a las políticas establecidas en el archivo de configuración. La cabecera de los mensajes rechazados se escribe en este archivo.

Registro de pánico

Si existe una entrada en el registro de pánico significa que Exim tiene un error, la mayoría de las veces el error es debido a una mala configuración en el archivo principal de Exim. Si todo va bien, el registro de pánico debe estar vacío.

3.4 QMAIL

Qmail es un Agente de Transporte de correo electrónico diseñado para funcionar en sistemas tipo Unix, fue escrito por Daniel J. Bernstein, en ese entonces un estudiante graduado en matemáticas por parte de la universidad de Berkeley en California. La primera aparición de qmail fue en enero de 2004 como una versión beta, la versión actual de qmail (1.03) fue lanzada en junio de 1998.

Qmail fue diseñado teniendo la seguridad como principal objetivo ya que el autor consideró que sendmail era un sistema de correo inseguro, por lo cual se dio a la tarea de construir un sistema de correo con características similares a sendmail pero mucho más seguro.

El software qmail tiene todas las funciones de un MTA, puede manejar el servicio de correo a través de SMTP, maneja cola de correos, entrega local de mensajes, soporta alias, listas de correo, usuarios virtuales, dominios virtuales, además incluye un servidor POP3 como soporte para los principales MUAs.

3.4.1 ARQUITECTURA DE QMAIL

Qmail es un sistema modular. Cada programa de qmail realiza una tarea en específico. Como resultado de lo anterior, los programas son pequeños, más simples y menos inseguros comparados con los programas de Sendmail. Para reforzar la seguridad del sistema de correo, los módulos de qmail tienen diferentes privilegios y no confían unos en otros, con lo que intercambian la mínima información posible para realizar sus tareas. Qmail sólo usa un programa que debe correr con el permiso setuid activado. Además sólo dos módulos de qmail tienen privilegios de administrador.

Los módulos que conforman a qmail son siete, a continuación se muestra su nombre y función.

qmail-smtpd	Este modulo acepta o rechaza mensajes a través del protocolo SMTP.
qmail-inject	Permite poner un mensaje en la cola a través del módulo qmail-queue
qmail-queue	Este módulo se encarga de poner

	mensajes en la cola.
qmail-rspawn/qmail-remote	Se encarga de manejar entregas de mensajes remotas.
qmail-lspawn/qmail-local	Gestiona entregas de mensajes locales
qmail-send	Procesa los archivo de la cola
qmail-clean	Limpia los archivo que se encuentran almacenados en la cola.

3.4.1.1 ESTRUCTURA DE ARCHIVOS QMAIL

Para que qmail funcione de manera adecuada se basa en ciertos archivos que permiten definir su comportamiento. Al igual que con los módulos, las diferentes características de qmail se mantienen en directorios diferentes formando de esta manera un pequeño sistema de archivos. El nodo raíz del sistema de archivos de qmail se encuentra en /var/qmail y aunque se puede cambiar en tiempo de instalación se recomienda mantenerlo en el lugar estándar. A partir del nodo raíz se cuelgan ciertos directorios que contienen archivos útiles en el funcionamiento de qmail, los directorios así como su contenido se describen en la siguiente tabla.

alias	En este apartado se encuentran los archivos de alias del sistema de correo.
bin	Archivos binarios y scripts, son programas para dar órdenes a los diferentes elementos que conforman qmail.
boot	Archivos de inicio del sistema de correo qmail
control	En este directorio se localizan los archivos de configuración que gobiernan el comportamiento de qmail
doc	En esta sección se localiza la documentación de qmail,

	exceptuando las páginas de manual.
man	Páginas de manual.
queue	Se encuentran mensajes no enviados.
users	La tabla de usuarios de qmail (opcional)

3.4.2 ARCHIVOS DE CONFIGURACIÓN DE QMAIL

Como se mencionó, la configuración de qmail se encuentra mayormente en los archivos que se encuentran en el directorio /var/qmail/control. Cada archivo dentro de este directorio contiene el valor o los valores que definen una sola característica en el comportamiento de qmail. Todos los archivos de configuración son opcionales excepto uno, su nombre es “me”. El archivo “me” contiene el nombre completo del equipo que actuará como servidor de correo.

Los archivos de configuración en qmail siguen algunas sencillas reglas en cuanto al formato que deben tener. A pesar de que se puede usar comentarios, los valores en cada archivo de configuración deben empezar en la primera línea, además no se permiten espacios en blanco, tabuladores o líneas en blanco extras.

Existen varios archivos de configuración en qmail, algunos de ellos tienen valores predeterminados y es precisamente el que usan si no se les asigna uno. Los principales archivos de configuración, la función que desempeñan así como los módulos que hacen uso de ellos son los siguientes.

Nombre	**Módulo que lo utiliza**	**Valor por default**	**Función**
badmailfrom	qmail-smtpd	Ninguno	Las direcciones que se encuentran en este archivo serán rechazadas por el servidor de correo. Cada una de las direcciones debe estar en una sola línea. Se pueden

			especificar nombres de dominio.
bouncefrom	qmail-send	MAILER-DAEMON	El valor de este archivo aparece como remitente en los correos que el servidor nos regresa por algún error encontrado. Si se cambia el valor de este archivo, se debe reiniciar el programa qmail-send para que los cambios surtan efecto.
bouncehost	qmail-send	El valor que contenga el archivo "me"	Este valor aparece en el nombre del host de los correos que el servidor nos regresa por un error en los mensajes. Se debe reiniciar el programa qmail-send para que los cambios surtan efecto.
concurrencyincoming	tcpserver	Ninguno	Limita el número de peticiones SMTP simultáneas que el servidor qmail aceptará.
concurrencylocal	qmail-send	10	Determina el máximo número de entregas simultáneas de manera local. El programa qmail-send debe ser reiniciado si se quiere refrescar el valor para este archivo.
concurrencyremote	qmail-send	20	Limita el máximo número de entregas remotas simultáneas.
databytes	qmail-smtpd		Establece un máximo en bytes para el tamaño de los mensajes recibidos vía SMTP. Un valor 0 indica

			que es ilimitado.
defaultdomain	qmail-inject	El valor que contenga el archivo “me”	Es el valor para el dominio que aparecerá en los mensajes enviados a partir del servidor que se está configurando.
locals	qmail-send	El valor que contenga el archivo “me”	Contiene una lista de dominios los cuales se tomarán como dominios locales (alias del dominio).
me	varios		Nombre completo del equipo que actuará como servidor de correo.
queuelifetime	qmail-send	1 semana	Contiene el tiempo máximo que un mensaje permanecerá en la cola de correos, en caso de que el tiempo expire se hace un último intento en la entrega del correo, si éste falla, el mensaje será regresado.
rcpthosts	qmail-smtpd	Ninguno	Lista de dominios de los cuales se aceptarán mensajes vía SMTP. Si este archivo se borra el servidor puede quedar como un equipo de relay abierto.
smtpgreeting	qmail-smtpd		Configura el banner que se presenta al inicio del diálogo SMTP.

timeoutconnect	qmail-remote	60 segundos	Tiempo que el programa qmail-remote espera para que se realice una conexión con una máquina remota.
timeoutremote	qmail-remote	1200 segundos	Tiempo que el programa qmail-remote espera por la respuesta a un comando enviado.

3.4.3 MANEJO DE QMAIL CON qmailctl

El script qmailctl proporciona una manera simple para controlar y monitorear a través de la línea de comandos el funcionamiento de qmail junto con algunos de sus módulos. Para utilizar qmailctl se debe proporcionar el nombre del script junto con un argumento que le indica la acción a realizar. La mayoría de las funciones realizadas por qmailctl requieren los privilegios del superusuario.

qmailctl stop

Este comando detiene los módulos qmail-send y qmail-smtpd. De esta manera ningún mensaje será entregado ya sea de manera local o remota y los mensajes almacenados en la cola no serán procesados. Además las peticiones al puerto SMTP serán rechazadas.

qmailctl restart

Esta orden detiene de manera momentánea el módulo smtpd mientras reinicia el módulo qmail-send. Pasar una señal de reinicio a qmailctl provoca que qmail-send vuelva a leer los archivos de control asociados a él.

qmailctl queue

Imprime un resumen del estado de la cola de correos así como el estado de cada uno de los mensajes que se encuentran en ella.

qmailctl stat

Imprime un resumen del estado actual de qmail, también indica si hay mensajes esperando a ser procesados

qmailctl help

Muestra un resumen de las funciones proporcionadas por el script llamado qmailctl.

3.5 POSTFIX

Postfix es un programa que actúa como MTA, maneja la entrega de mensajes entre servidores así como de manera local. Para realizar el trabajo de entrega-recepción de mensajes a través de la red, Postifx utiliza el protocolo SMTP. Para la entrega local, existe un agente en el programa de postfix que se encarga de depositar los mensajes en los buzones de los usuarios o en su defecto entrega los mensajes a un agente especializado en entregas locales, un MDA.

Postfix se diseño con la idea de reemplazar a sendmail. Postfix trata de eliminar la complejidad en el manejo de sendmail, también intenta solucionar los problemas de seguridad que sendmail tiene. Este sistema de correo fue escrito por Wietse Venema, un experto en seguridad conocido por desarrollar el paquete TCP Wrappers. La aparición de postfix como programa libre fue en diciembre de 1998, con el patrocinio de IBM.
Los principales objetivos en el diseño de Postfix son:

Eficiencia
Postfix muestra su poder bajo ambientes de alta carga.

Seguridad
Postfix asume que está siendo ejecutado bajo ambientes hostiles. Por lo cual emplea varias capas de defensa contra atacantes. El principio del mínimo privilegio es el que usa postfix en sus procesos. Asimismo los módulos que no se necesitan se pueden deshabilitar para reforzar la seguridad y simplificar la instalación.

Rendimiento
Uno de los puntos fuertes de Postfix es el rendimiento. De hecho, este sistema de correo usa técnicas para limitar el número de nuevos procesos y el número de accesos al sistema de archivos para procesar mensajes.

Flexibilidad
El sistema de correo Postfix está formado por varios programas

diferentes, lo que permite que se pueda optimizar cada una de las piezas del sistema a través de archivos de configuración fáciles de administrar.

Facilidad de Uso
Postfix es uno de los sistemas de correo más fáciles de instalar y de administrar. Sus archivos de configuración son fáciles de entender, y las variables tienen un nombre descriptivo. También proporciona comandos con parámetros fáciles de entender y recordar.

Compatibilidad con Sendmail
Postfix puede reemplazar a Sendmail ya que maneja muchas convenciones de este servidor de correo. El binario ejecutable "sendmail" se sustituye por una versión de Postfix que soporta casi todas las opciones de la línea de comandos que sendmail proporciona. Postfix al igual que sendmail puede manejar archivos de alias y archivos .forward.

Todas las similitudes entre Sendmail y Postfix son sólo en funcionalidad porque Postfix las implementa de un modo muy diferente a Sendmail.

3.5.1 ARQUITECTURA DE POSTFIX

Postfix tiene una arquitectura modular la cual es la base de su seguridad. Cada proceso se ejecuta con el mínimo privilegio para realizar su tarea, incluso los procesos que no se necesiten se pueden inhabilitar haciendo el sistema más simple y seguro. Los procesos de postfix dependen muy poco unos de otros, si un proceso se compromete es difícil que el impacto se propague a los demás procesos.

La mayoría de los programas que componen a Postfix son procesos que se ejecutan en segundo plano. Un demonio llamado master se inicia primero e invoca a la mayoría de los procesos restantes, esto lo hace conforme se van necesitando los demás procesos. Los demonios que son invocados por el programa master realizan su tarea y terminan. Sin embargo, el demonio master se mantiene en ejecución. El comportamiento del demonio master lo determinan sus archivos de configuración los cuales son: master.cf y main.cf.

3.5.2 ENTRADA DE MENSAJES EN POSTFIX

De modo general, la forma en que Postfix procesa los mensajes es el siguiente: primero recibe los mensajes, después los pone en la cola de

correos para finalmente entregarlos a su destino. Cada paso de este proceso es manejado por un conjunto de componentes diferentes de Postfix. Una vez que el mensaje ha sido recibido y se encuentra en la cola, el gestor de la cola de correos es quien se encarga de entregar el mensaje a un agente especializado para el depósito del correo en su destino.
Postfix puede recibir los mensajes por distintos medios:

- Un mensaje puede entrar al sistema de correo Postfix de manera local, dicho de otra forma, es un mensaje enviado por un usuario local.
- Un mensaje proveniente de la red puede ser procesado por Postfix.
- Un mensaje que ya ha sido aceptado por Postfix por cualquiera de los métodos ya comentados y es preparado para su reenvío a otra dirección.
- Un mensaje puede ser generado por el mismo sistema de correo cuando es necesario enviar una notificación de errores en la entrega.

3.5.2.1 ENTREGA DE MENSAJES LOCALES

El programa administrador de la cola, aparte de gestionar los mensajes que llegan a la cola se encarga de notificar al componente adecuado cuando tiene algún trabajo por hacer.

Para procesar un mensaje de manera local se siguen los siguientes pasos: El mensaje se deposita en el directorio maildrop a través del comando postdrop. El demonio pickup recoge el mensaje de la cola y se lo manda al demonio cleanup. En ocasiones los mensajes llegan a este punto sin el formato correcto para ser un correo válido, es por ello que el demonio cleanup junto con el demonio trivial-rewrite revisan los mensajes y en caso de ser necesario agregan cabeceras a los mensajes, convierten las direcciones a un formato adecuado e incluso traducen direcciones basándose en tablas canónicas o virtuales. Una vez que el demonio cleanup termina su trabajo, notifica al programa gestor de la cola que ha cedido el mensaje al módulo incoming. Es en este punto cuando el gestor de la cola invoca al agente especializado en entregas locales para que se haga cargo del mensaje.

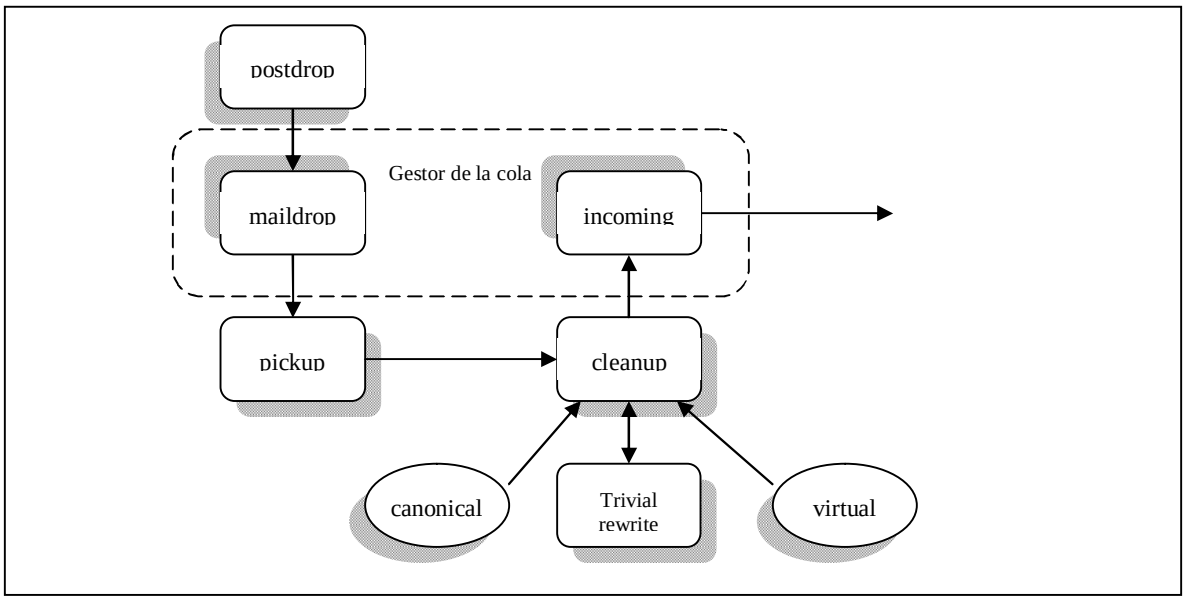


Figura 3.1 Camino que sigue un mensaje local cuando entra al sistema Postfix.

3.5.2.2 ENTREGA DE MENSAJES PROVENIENTES DE LA RED

Los mensajes que entran provenientes de la red son aceptados por el demonio `smtpd`. Este programa hace una revisión general de los mensajes entrantes, además está habilitado para permitir o denegar el relay. Después que el demonio `smtpd` termina su trabajo cede los mensajes al demonio `cleanup` quien hace una revisión más exhaustiva de los correos y deposita los mensajes en el módulo `incoming`. En este punto, el programa gestor de la cola invoca al agente especializado para la entrega de mensajes para que se haga cargo de los mismos.

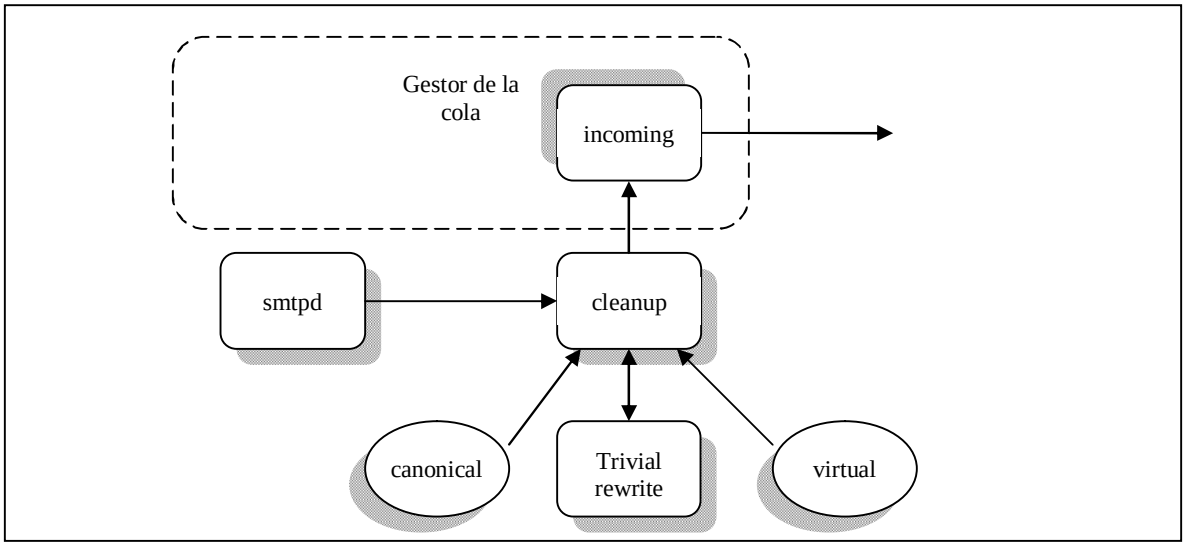


Figura 3.2 Flujo de los mensajes provenientes de la red.

3.5.3 CONFIGURACIÓN Y ADMINISTRACIÓN DE POSTFIX

Cuando se instala Postfix se tiene una configuración por default que funciona para usuarios que solo requieren enviar y recibir correo. Sin embargo si lo que se desea es explotar las funcionalidades de Postfix para tener un sistema personalizado, se deben modificar los parámetros en los principales archivos de configuración de Postfix.
En una instalación común los principales directorios de postfix se encuentran en los siguientes lugares:

/etc/postfix
Este directorio contiene los archivos de configuración de Postfix, así como las tablas de traducción de direcciones.
/usr/libexec/postfix
este directorio alberga los demonios de Postfix

/var/spool/postfix
Contiene los archivos relacionados con la cola de correos.

/usr/sbin
En este lugar se localizan los comandos para controlar Postfix.

En una instalación de postfix se debe crear un usuario y grupo para que se hagan cargo del sistema de correo, usualmente son `postfix` y `postdrop` para el usuario y grupo respectivamente.

Antes de iniciar Postfix por primera vez se necesita verificar algunos datos para que funcione sin problemas. El primero de ellos es un parámetro de configuración llamado `myhostname`, este parámetro define la identidad del equipo donde se instaló Postfix, es decir, es el nombre y dominio de la máquina que se usará como sistema de correo. Una vez que postfix conoce el nombre completo del equipo, lo utiliza para establecer otras variables importantes, por ejemplo: `mydomain`. En caso de que el usuario que instala Postfix no proporcione un nombre de host, Postfix trata de encontrarlo con el comando `hostname`. Si este comando no arroja el nombre completo del equipo, Postfix no funcionará de manera correcta. Para agregar el nombre del equipo de forma manual, se puede proceder de dos maneras: como primera opción se puede utilizar un editor para cambiar de manera directa el valor de la variables en el archivo `main.cf`. La segunda forma es utilizando el comando `postconf` con el parámetro “-e” que permite editar el archivo de configuración de Postfix con el parámetro y valor proporcionados desde la línea de comandos.

```
# postconf -e myhostname=alfredo.ingenieria.mx
```

El segundo punto a tomar en cuenta antes de iniciar Postfix por primera vez, consiste en revisar el formato del archivo de alias de correo. Debe existir un archivo de texto de alias para que lo pueda modificar el usuario y debe existir un archivo de alias binario para que lo pueda leer el sistema de correo. Cada vez que se modifica el archivo de alias se debe ejecutar el comando newaliases para actualizar el archivo de alias binario.

3.5.3.1 ARCHIVOS DE CONFIGURACIÓN

Los dos archivos de configuración más importantes en Postfix se encuentran en /etc/postfix y se llaman master.cf y main.cf. Estos dos archivos deben pertenecer al usuario root y tener permisos de escritura sólo para él, aunque el resto de los usuarios los puedan leer.

El archivo master determina la forma en que se comportarán los módulos que hacen labores de transporte en el sistema de correo.

Formato del archivo main.cf

El archivo main.cf es el corazón en la configuración de Postfix. Casi todas las opciones para las variables que maneja Postfix se encuentran dentro de main.cf. Este archivo se puede editar con el comando postconf o con un simple editor de texto como vi o emacs.

En el archivo de configuración de postfix se pueden escribir líneas en blanco, comentarios y líneas que asignan valores a los parámetros. Estos elementos siguen un determinado formato.

- Los comentarios inician con el carácter "#" y continúan hasta el fin de la línea.
- La líneas en blanco al igual que los comentario son ignorados por Postfix.
- Para definir un parámetro éste debe empezar en la primera columna de la línea, debe contener el nombre del parámetro seguido del signo "=" y a continuación el valor.
- No se puede tener un comentario en la misma línea en que se tiene un parámetro.
- No se recomienda usar comillas dobles alrededor del valor de las variables, en caso de encontrar el valor de una variable con

comillas dobles, Postfix las considerará como parte del valor de dicha variable.
- Una línea que inicia con espacios o tabulaciones se considera una continuación de la línea anterior.
- Se puede definir el valor de una variable a través del nombre de otra variable anteponiendo el signo de “$”.
- Muchos parámetros pueden tener más de un valor. Si existen varios valores para una variable éstos deben separarse por comas, tabulaciones, espacios o líneas nuevas.

Siempre que se modifica el archivo de configuración main.cf, se debe pasar el parámetro reload al programa postfix para que los cambios surtan efecto.

3.6 MDA

Agentes especializados para la entrega de los mensajes en los buzones de cada uno de los usuarios.

3.6.1 PROCMAIL

Procmail es un Agente de Entrega de Correo Electrónico (MDA) que puede proporcionar opciones de filtrado de mensajes de acuerdo al remitente, asunto, tamaño del mensaje, palabras clave dentro del mensaje, etc. Aparte de lo anterior, procmail puede redireccionar mensajes hacia carpetas o a otras direcciones, también permite llamar a programas externos como pueden ser un antivirus o un analizador de Spam. Todo lo anterior lo hace procmail gracias a reglas definidas ya sea por el administrador del equipo o por un usuario con cuenta en el sistema.

Procmail es uno de los MDA más utilizados ya que se puede integrar con la mayoría de los MTA de una forma sencilla. Fue desarrollado por Stephen Van den Berg y actualmente es mantenido por Philip Guenther

3.6.1.1 REGLAS DE PROCMAIL

Para procesar los correos procmail se basa en un archivo de reglas que tiene un formato definido. El archivo de reglas generalmente tiene tres secciones, la primera de ellas determina las variables de entorno a utilizar, la segunda sección establece las reglas que debe cumplir el correo para ser procesado y la tercera parte contiene alguna acción a realizar con el correo coincidente.

La primera sección de un archivo de reglas contiene las variables de entorno, las más comunes son las siguientes:

MAILDIR
Indica el directorio donde se almacenan los archivos que contienen los correos. Usualmente esta variable apunta a /var/spool/mail o a $HOME/mail

LOGFILE
Especifica el archivo donde procmail registra sus acciones.

FORMAIL
Establece la ruta donde se encuentra el programa formail. Este programa se distribuye junto a procmail y sirve para modificar las cabeceras de los mensajes o reformatear un mensaje antes de enviarlo o almacenarlo.

DEFAULT
En caso de que ninguna de las reglas se aplique al mensaje, éste se almacenará en el archivo indicado por la variable DEFAULT.

La sección intermedia en un archivo procmail contiene las reglas para el procesado de los mensajes.

La sintaxis general de una regla es la siguiente:

```
:0 [opciones] [ : [fichero de exclusión] ]
* condicion 1
* condicion 2
    .

    .
    .
* condición N
comando
```

En primer lugar cada regla comienza por un :0, a continuación aparecen las opciones que pueden ser:

H La condición se aplica a la cabecera del mensaje.
B La condición se busca en el cuerpo del mensaje.
D Al analizar la condición se distingue entre mayúsculas y minúsculas.

A Esta regla se ejecutará únicamente si su antecesora lo hizo.
a Igual que la anterior, con la salvedad de que la ejecución de la regla anterior debió realizarse sin errores.
E Esta regla se ejecutará si la anterior no lo hizo.
e Esta regla se ejecutará si se intento ejecutar la regla anterior pero hubo algún error.
h La cabecera se pasa al comando.
b El cuerpo del mensaje se pasa al comando.
f El comando se considerará como un filtro.
c Genera un *copia* del mensaje. Al ejecutar una regla que da el mensaje por entregado con este flag, se consigue que el mensaje no se dé por entregado y se puedan ejecutar otras reglas a continuación de esta.
w Espera a que el comando se ejecute para recibir su código de salida.
W Igual que el anterior pero en caso de error no emite ningún mensaje.
i Ignora los posibles errores de escritura.
r Escribe el mensaje tal y como esté. No comprueba que termine en una línea en blanco que sería lo correcto.

De no indicarse nada, se comparará la condición de la regla con la cabecera del mensaje (opción H). Al comando se le pasará tanto la cabecera del mensaje como su cuerpo (opciones h y b). No se hará distinción entre mayúsculas y minúsculas.

Tras el :0 y las posibles opciones puede aparecer opcionalmente un segundo ":", de hacerlo se estará indicando que el fichero destino donde se escriba el mensaje debe bloquearse para que dos procesos no escriban a la vez sobre el fichero. Opcionalmente se puede indicar el fichero de exclusión que se usará para realizar el bloqueo.

A continuación vienen las condiciones, una por línea y precedidas por un *. En las condiciones generalmente se usan *expresiones regulares* para intentar encontrar cadenas de texto dentro de la cabecera o del cuerpo del mensaje. Las expresiones regulares usan los siguientes símbolos:

^ Comienzo de la línea.
$ Final de la línea.
. Cualquier carácter excepto un salto de línea.
***** Cero o más veces.
+ Una o más veces.
? Cero o una vez.
[a-z] Rango de caracteres, en este caso de la 'a' a la 'z'.
[^a-z] Cualquier carácter que no esté en el rango de la 'a' a la 'z'.
a|b La 'a' o la 'b'

COMANDOS

Tras cada condición se especifica un comando para que sea ejecutado si la regla se cumple. Distinguimos principalmente cuatro comandos básicos:

Archivo: Si en la sección de comandos aparece el nombre de un archivo procmail agrega el mensaje al final del archivo.

Directorio: Hace que procmail guarde el mensaje en el directorio con un nombre propio no repetido.

!direccion@email: Mediante el carácter '!' podemos enviar el mensaje a la dirección de correo especificada.

|programa: El carácter '|' permite ejecutar un programa o comando de linux.

Dentro de un archivo de reglas de procmail es importante mencionar que aquellos caracteres que tengan un significado especial para PROCMAIL deberán ser "escapados" mediante la barra inversa.

```
# ejemplo
MAILDIR=$/var/spool/mail
DEFAULT=$MAILDIR/jose
LOGFILE=$MAILDIR/log


:0
* ^From:.*usuario@no.admitir
/dev/null
```

Con la regla anterior todos los correos provenientes de la dirección usuario@no.admitir serán destinados a /dev/null, en otras palabras, serán desechados.

3.6.2 MAILDROP

Maildrop es un filtro de mensajes y agente de entrega de correo que se distribuye con el servidor de correo Courier, aunque también puede ser instalado por separado para integrarlo con otros servidores de correo.

Maildrop es un software desarrollado con C++ y es más grande en código que procmail, debido a esto tiene mucho más opciones de procesado que procmail.

3.6.2.1 ARCHIVO DE FILTROS

Maildrop puede filtrar un mensaje a partir de reglas definidas en un archivo. Las reglas se definen en un lenguaje de filtros estructurado y se basa en la coincidencia de patrones. En caso de que no exista un archivo de filtros, maildrop simplemente entrega el mensaje sin hacer algún procesamiento adicional.

VARIABLES

Maildrop utiliza variables para manipular mensajes. Para declarar variables en el archivo de filtros de maildrop se utiliza el nombre de la variable en mayúsculas (normalmente) seguido del signo igual y el valor de la variable entre comillas dobles. Cuando se quiere acceder al valor de una variable se le antepone el signo $ al nombre de dicha variable.

Existen variables definidas automáticamente por maildrop, éstas se pueden cambiar por el administrador del sistema.

DEFAULT

Se refiere al buzón por default donde se entregarán los mensajes. Si el archivo de filtros no específica un buzón, el mensaje será entregado a el buzón definido en la variable DEFAULT.

FROM

Usualmente hace referencia a la dirección de donde proviene el mensaje.

HOME

El directorio HOME del usuario que ejecuta maildrop.

HOSTNAME

Nombre de la máquina donde se ejecuta maildrop.

PATH

Ruta de los comandos que se ejecutarán en el archivo de filtros.

SHELL

El shell por default que ejecuta los comandos del sistema.

LINES

Número de líneas del mensaje procesado actualmente.

SIZE

Tamaño en bytes del mensaje procesado

SUSTITUCIÓN DE COMANDOS

El texto encerrado entre acentos invertidos será interpretado como comandos del shell por maildrop. Los comandos de shell se ejecutan como proceso hijo.
DIR=`pwd`

Ejecuta el comando pwd y el valor de sus salida se lo asigna a la variable DIR, en este caso específico el valor de DIR será el directorio actual.

BÚSQUEDA DE PATRONES

La sintaxis para la búsqueda de patrones es parecida a la del comando grep. Una búsqueda de patrones tiene la siguiente sintaxis en el archivo de filtros:

/*patrón*/:*opciones*

patrón especifica el texto a buscar en el mensaje. Ciertas letras tienen un significado especial cuando se hace una búsqueda de patrones.

h

Compara el patrón con la cabecera del mensaje.

b

Compara el patrón con el cuerpo del mensaje.

D

Hace la comparación distinguiendo mayúsculas y minúsculas, normalmente la comparación de patrones no es sensible a mayúsculas o minúsculas.

En caso de que no se especifique ‘h’ o ‘b’, el patrón se compara sólo con la cabecera del mensaje. Para especificar que la comparación se debe hacer con el mensaje completo se deben poner ambas letras, ‘h’ y ‘b’. Cuando la comparación tiene éxito, la línea coincidente se asigna a la variable MATCH para su posterior procesamiento.

Expresiones

A pesar que maildrop evalúa expresiones numéricas, los resultados de esas expresiones se almacenan como literales de texto.

Operadores

En el archivo de filtros podemos trabajar con operadores, los siguientes son los operadores disponibles en maildrop, se escriben de acuerdo al orden de precedencia que tienen.

```
||
&&
<  <=  >  >=  ==  !=  lt  le  gt  ge  eq  ne
|
&
+  -
*  /
=~ /pattern/
/pattern/  !  ~  function()
```

REGISTRO DE PROCESO

El registro de procesamiento de mensajes esta deshabilitado por default en maildrop. La sentencia *logfile* permite especificar un archivo donde se registrará el procesamiento de los mensajes. Si el archivo pasado como argumento a la sentencia logfile existe, entonces maildrop agrega el nuevo procesamiento a ese mismo archivo.

BUCLES Y CONDICIONES

Para buscar y procesar patrones de una forma más eficiente, maildrop proporciona sentencias condicionales e iterativas.

foreach

```
foreach /patrón/:opciones
{
    ...
}
```

```
foreach (expressión) =~ /patrón/:opciones
{
    ...
}
```

La expresión **foreach** ejecuta un bloque de sentencias por cada ocurrencia de un patrón en el mensaje.

while

```
while (expression)
{
    ...
}
```

Con esta sentencia se pueden crear ciclos. La expresión entre paréntesis se evalúa en cada iteración del bucle, mientras sea verdadero el bucle se ejecuta, en el momento que la expresión resulte falsa el ciclo termina

if

```
if (expression)
{
    ...
}
else
{
    ...
}
```

La sentencia if permite tomar decisiones respecto a la ejecución de determinadas sentencias. Si la expresión es verdadera sólo se ejecutan las sentencias que se encuentran inmediatamente después del if. En caso de que la expresión resulte falsa sólo se ejecutan las sentencias que le siguen a la palabra “else”.

exit

La sentencia **exit** termina de forma inmediata el filtrado de un mensaje. Cuando se utiliza “exit”, maildrop termina la operación de filtrado sin la entrega del mensaje. La sentencia exit se usa por lo regular cuando **maildrop** se ejecuta en modo embebido y cuando las instrucciones de entrega del mensaje no están permitidas.

3.7 MUA

Al principio del capítulo se definió a un Agente de Correo de Usuario (MUA) como el programa encargado de proporcionar al usuario una forma fácil de escribir, leer y almacenar correos electrónicos. Existen varios tipos de MUAs en el mercado, algunos de ellos ofrecen una interfaz gráfica pero muchos ofrecen sus funciones a través de texto.

Ejemplos de MUAs son: Outlook, PINE, Thunderbird, Eudora, Evolution, Kmail, entro otros. A continuación se mencionan dos MUAs ampliamente utilizados en el correo electrónico.

3.7.1 PINE

Pine es un MUA que permite leer, enviar y gestionar correo electrónico con una interfaz de texto, se encuentra disponibles para sistemas tipo Unix y existen versiones para Windows. Fue desarrollado en la Universidad de Washington, su primera aparición se remonta a 1989.

Pine maneja muchas opciones y tiene manuales en línea si es que se desea saber acerca del uso del programa. En esta sección sólo se listarán las características con que cuenta Pine.

3.7.1.1 CARACTERÍSTICAS DE PINE

- Ayuda. Contiene Información acerca del uso del programa.

- Índice de Mensajes. Muestra un resumen de los mensajes que se encuentran en el buzón. El resumen incluye el estado, remitente, tamaño, fecha y asunto de cada uno de los mensajes.

- Comandos. Pine proporciona comandos para gestionar los mensajes, por ejemplo permite reenviar, responder, salvar, exportar, imprimir, borrar y buscar mensajes.

- Compositor de Mensajes. Es un editor de fácil uso para escribir mensajes de correo. Ofrece una interfaz con ciertos comandos para agilizar la escritura del mensaje, además proporciona acceso directo a la libreta de direcciones.

- Libreta de Direcciones. Permite mantener direcciones complejas y una larga lista de direcciones de correo a través de alias.

- Mensajes Adjuntos. Maneja la especificación MIME para archivos adjuntos lo que significa que permite enviar/recibir archivos binarios, gráficos, sonido, además de texto.

- Comandos para el manejo de Carpetas. Se utilizan para crear, borrar o renombrar carpetas de mensajes. Las carpetas pueden ser locales o remotas

- Acceso a carpetas y mensajes remotos. Se puede tener acceso a mensajes electrónicos que residen en otra máquina a través del protocolo IMAP.

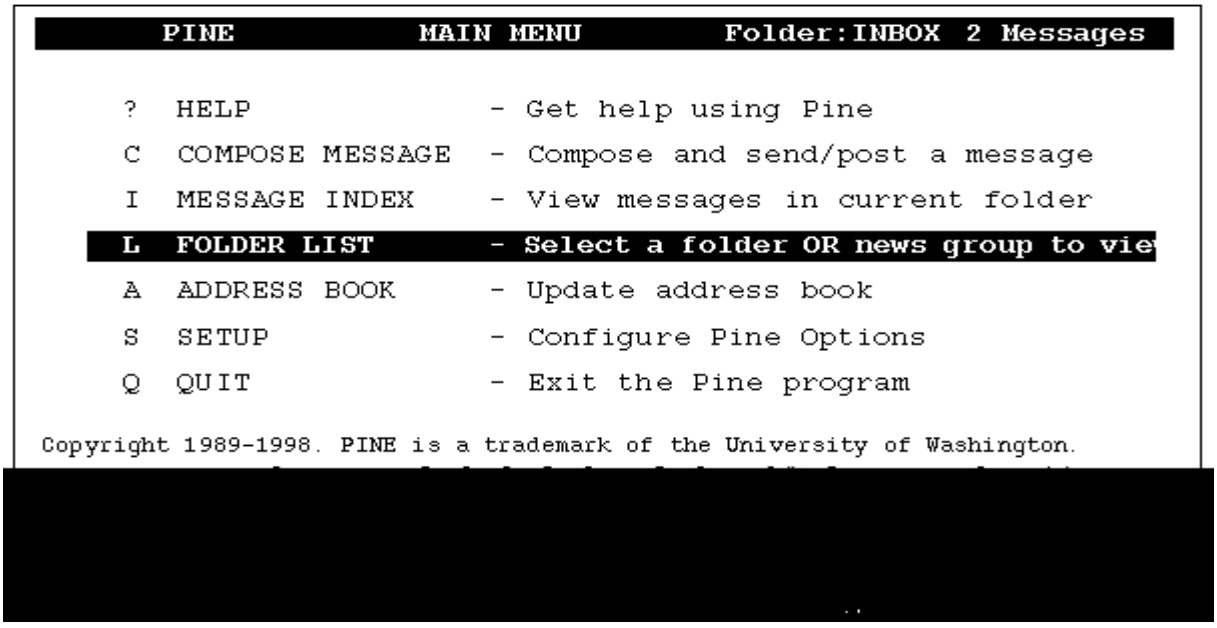


Figura 3.3 Pantalla del menú principal de Pine.

Escribir mensajes en Pine es muy fácil ya que proporciona palabras que sugieren la información que debe llevar un mensaje, incluso en la parte inferior de las distintas pantallas de Pine se muestran los comandos para cada opción que se quiera manejar.

3.7.2 OUTLOOK EXPRESS

Outlook Express es un cliente de correo electrónico y de noticias derivado de Outlook. Fue creado por Microsoft para distribuirlo con su sistema operativo Windows. Outlook Express proporciona un entorno gráfico para el manejo de correo electrónico, tiene una barra de comandos que facilita las operaciones relacionadas con el correo. Entre sus principales características destacan:

- Administración de diferentes cuentas de correo.
- Envío/recepción de correo en distintos formatos (texto, scripts, sonido, multimedia, etc.)
- Conexión a servidores remotos por medio del protocolo POP3.
- Libreta de Direcciones.
- Facilidades de Importación de datos (mensajes, libretas de direcciones, cuentas, etc.) de otras cuentas o de otros clientes de correo.
- Entorno amigable.
- Compatibilidad con HTML.
- Administración de carpetas (bandejas de entrada, salida, etc.).
- Reglas de filtrado.

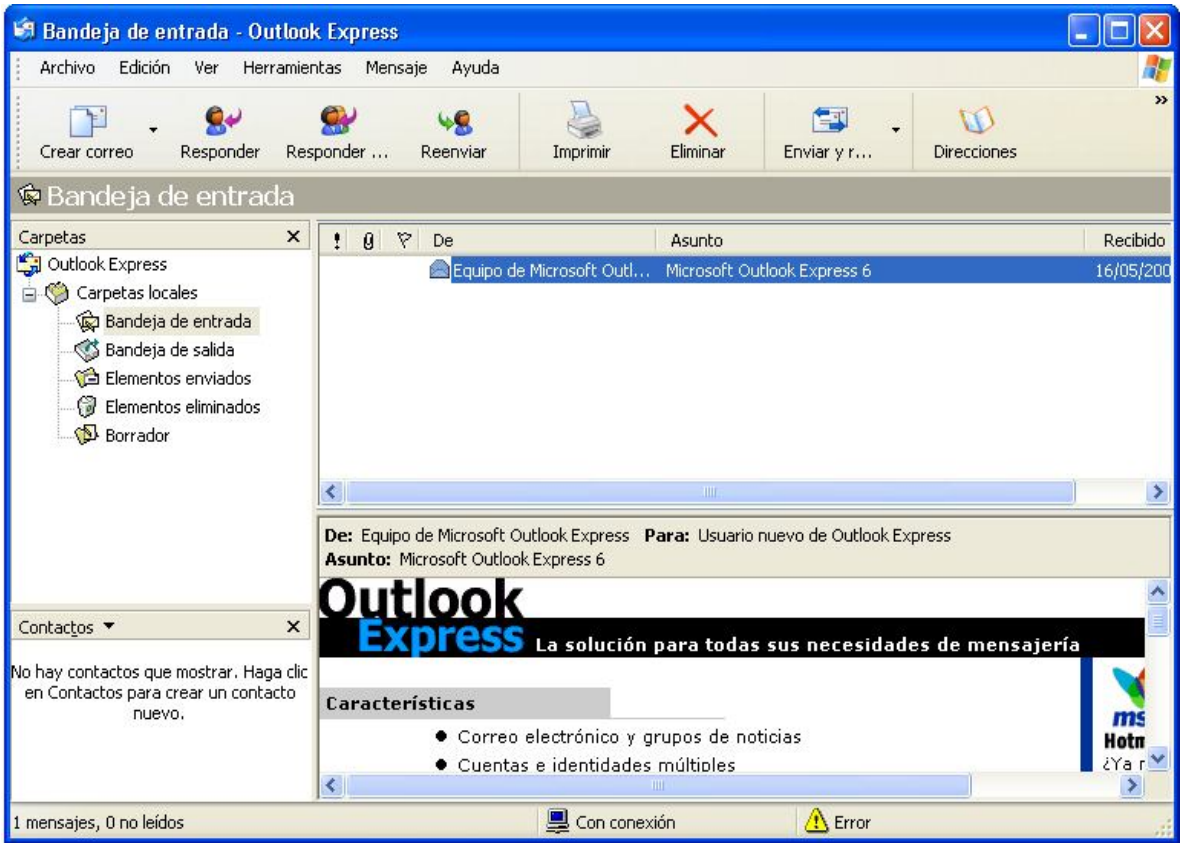


Figura 3.4 Bandeja de entrada de Outlook Express

3.8 VARIEDAD DE CLIENTES DE CORREO

Existen varios programas que actúan como clientes de correo. La adopción de cada uno depende de las preferencias del usuario. Las características de los clientes abarcan desde el sistema operativo que lo soporta, las características, el costo, la facilidad de uso, etc.

A continuación se muestra una tabla resumiendo las características de algunos sistemas cliente de correo.

Cliente	Sistema Operativo	Licencia	Seguridad	Fabricante	Características
Eudora	-Windows -Mac OS	Propietaria	-POP sobre SSL y TLS -IMAP sobre SSL y TLS -SMTP sobre SSL y TLS	Qualcomm	-Bloqueo de Imágenes Filtro Antispam -Correo HTML -Interfaz gráfica
Evolution	-Linux -Windows -Mac OS	GPL	-POP sobre SSL y TLS -IMAP sobre SSL y TLS -SMTP sobre SSL y TLS -SASL -Filtro AntiSpam -PGP -S/MIME	Gnome, Novell	-Bloqueo de Imágenes Filtro Antispam -Correo HTML -Interfaz gráfica -Conjunto de caracteres variado -Intercambio de datos con dispositivos móviles -Calendario -Libreta de direcciones -Búsquedas de correo

Kmail	Linux	GNU GPL	-PGP -Filtros AntiSpam -POP sobre SSL y TLS -IMAP sobre SSL y TLS -SMTP sobre SSL y TLS DIGEST-MD5	KDE	-Interfaz gráfica - Correos en HTML - Búsquedas -Importación de datos de otros clientes -Verificación de ortografía -Conjunto de caracteres para escribir correo en cualquier Idioma -Libreta de direcciones -Compresor de adjuntos
Outlook Express		Propietaria	-SMTP Auth -POP sobre SSL -IMAP sobre SSL	Microsoft	-Contenidos Activos DHTML y ActiveX -Libreta de direcciones -Interfaz Gráfica -Bloqueo de Imágenes -Correo HTML
Pine	GPL	Licencia PINE	-MD5 -Filtros	U. de Washington	-Interfaz basada en texto -Libreta de direcciones
Mozilla Thunderbird	-Linux -Windows -Mac OS	GPL	-Filtro contra Phishing -PGP -Filtros contra correo	Grupo Mozilla	-Interfaz gráfica -Variedad de Complementos -Marcadores de Mensajes -Bloqueo de

			basura -S/MIME		Imágenes -Búsquedas -Libreta de direcciones

Tabla 3.1 Variedad de clientes de correo.

CAPÍTULO 4

SEGURIDAD

4.1 CLAVES DE LA SEGURIDAD DEL CORREO

La seguridad de las comunicaciones no es un problema nuevo. En el correo convencional, se pierdan cartas y se entregan en la dirección equivocada. También existe el riesgo de que el mensaje sea interceptado por algún trabajador malicioso si se sospecha que contiene algo de valor. Los sistemas de correo electrónico tienen problemas similares. Los mensajes se pierden. Pueden llegar a la dirección equivocada o se pueden interceptar durante el envío. Es posible que, al recibir un mensaje, no pueda identificar al emisor. También, la posibilidad de anexar mensajes trae consigo el riesgo de que éstos transmitan algún virus, o malware (agente malicioso).

Las claves de la seguridad del correo electrónico se resumen en las siglas CIA.

Confidencialidad: garantizar que terceras partes no tengan acceso al contenido de los mensajes.

Integridad: asegurar que el mensaje llegue al destinatario indicado y que el contenido no sea alterado, accidental o deliberadamente, durante la transmisión.

Autenticación: Garantizar que el mensaje procede realmente de quien figura como emisor y que éste no pueda negar el hecho de haberlo enviado.

Los servidores y los clientes de correo electrónico son objetivo de múltiples ataques informáticos debido a que intercambian datos con redes no confiables. Además, tanto los clientes como los servidores de correo se han utilizado como medios para insertar malware dentro de las máquinas y propagarlo mediante mensaje electrónicos. Es por ello que servidores y clientes de correo así como la infraestructura de red que los soportan deben ser protegidos.
Ejemplos de algunos problemas con la seguridad del correo:

- Los correos pueden ser usados como un medio para realizar ataques de ingeniería social.

- Errores en la implantación del servidor pueden ser usados como medios parta comprometer el sistema y la red. Por ejemplo, se puede ganar acceso a carpetas o archivos privados o ejecutar

comandos en el servidor, incluso, en algunos casos hasta la instalación de software.

- Ataques de denegación de servicio (DoS) pueden ser dirigidos hacia servidores de correo.

- Información privada del servidor de correo puede ser leída por usuarios no autorizados o modificada de manera no autorizada.

- Información sensible transmitida sin cifrar entre el servidor y el cliente puede ser interceptada. Los protocolos más utilizados para el intercambio de mensajes por default envían nombres de usuario, contraseñas y mensajes sin cifrar.

- Se puede realizar ataques a organizaciones externas a partir de un sistema de correo comprometido.

- Una mala configuración en el servicio de correo puede permitir que el sistema sea utilizado como distribuidor de correo basura (spam).

- Los usuarios pueden mandar información privada de la empresa, contenido inapropiado o contenido con derechos de autor que expone a la organización a problemas legales.

Debido a lo anterior, las organizaciones deben implementar algunos procedimientos para mantener seguro un servidor de correo.

Procedimientos adecuados son esenciales para asegurar los recursos de un servidor de correo. Las prácticas de seguridad agrupan la identificación de la información del sistema, documentación, implementación de políticas, estándares, procedimientos y guías que ayuden a asegurar la confidencialidad, integridad, y disponibilidad de los recursos proporcionados por el servidor
Algunas de las prácticas se resumen a continuación:

- Implementación de políticas de seguridad

- Cifrado y firmas digitales en los mensajes

- Control, y manejo de la configuración del servidor
- Administración y control de riesgos.

- Conocimiento del sistema.
- Contingencia, disponibilidad y un plan de recuperación de desastres.
- Capacitación.

El primer paso para asegurar un sistema de correo es asegurar el sistema operativo sobre el cual está instalado. Otro punto que se debe considerar en la puesta en marcha de un servidor de correo es la implementación de mecanismos criptográficos para proteger los mensajes. La mayoría de los protocolos de correo envían los datos de autenticación de los usuarios así como los mensajes en texto claro. De esta manera, un atacante puede interceptar el correo y fácilmente alterar el contenido de él.

Mantener la seguridad de un sistema de correo es un proceso continuo en los que se requiere esfuerzo, monitoreo del sistema, recursos de la organización, etc.
La seguridad del Servidor de correo contempla los siguientes pasos:

- Configurar, proteger y analizar los archivos de registro (logs) bitácoras.
- Respaldar la información de manera frecuente.
- Protegerse contra malware (virus, gusanos, caballos de troya)
- Establecer y seguir los procedimientos establecidos para una recuperación en caso de que el sistema sea comprometido
- Probar e instalar actualizaciones en momentos adecuados.
- Probar la seguridad del sistema de forma periódica.
- Actualizar las políticas.

4.2 POLÍTICAS

Una política es un conjunto de normas a través de las cuales una empresa hace saber a sus empleados cómo quiere que traten un determinado asunto, en este caso el correo.

El correo electrónico es una herramienta de trabajo que se está extendiendo a una velocidad tal que es difícil esperar que los empleados conozcan y entiendan todos los riesgos y ventajas del sistema. Si la organización tiene una política al respecto, proporciona a los usuarios un marco de actuación y les permite dar respuesta a cuestiones que generalmente desconocen.

Antes de crear una política debe entender cuáles son los riesgos y pensar qué normas puede implantar para protegerse contra ellos.

Conocer los riesgos

Una infección de virus que se aloje en un archivo anexo puede llevar, a su vez, a contagiar a otros.

Saturación de redes internas por envío de gran volumen de mensajes carentes de interés o mensajes con anexos muy largos.

El envío de mensajes con contenido difamador, machista, racista o potencialmente ilegal.

La posibilidad de dar a conocer información confidencial de la empresa.

La posibilidad de revelar datos personales.

Potenciar las ventajas del correo electrónico

Aunque las políticas sirven para controlar los riesgos, también han de tener en cuenta el beneficio que el correo aporta a la organización. Una buena política le ayudará a potenciar las ventajas del correo electrónico. El correo electrónico puede ser de ayuda para:

- Mejorar la comunicación interna.
- Establecer una relación más cercana e interactiva con los clientes.
- Mejorar la productividad.
- Reducir los costes.

No se debe considerar una política exclusivamente como un medio de frenar aquello que no desea que ocurra sino también como una forma de ayudarle a lograr lo que quiere.

Una buena política ayuda a la organización a sacar el mayor partido de su inversión. No existe una política adecuada para todas las organizaciones. Cada empresa tiene características únicas derivadas de su cultura y de su funcionamiento y la política debe reflejar su carácter exclusivo.

Puntos que debe cubrir una política:

- Riegos y beneficios del e-mail para la empresa.
- Seguridad y confidencialidad.
- Estatus legal de los mensajes.
- Diferencias entre el uso de correo interno y externo.
- Información que debe incluirse en los mensajes externos.
- El uso del correo para asuntos personales.
- Gestión (almacenamiento, recuperación y borrado de los mensajes enviados y recibidos.
- Corrección y tono de los mensajes.
- Referencias a las normas de la empresa y directrices practicas.
- Responsabilidad personal de cada usuario de sistema.
- Seguimiento y control del sistema de correo.
- Normas para el tratamiento de los mensajes inapropiados recibidos.
- Posibles consecuencias de ignorar la política.

Características de una buena política

- Está bien escrita.
- Es práctica.
- Es flexible: Se puede actualizar con facilidad.
- Es de fácil acceso: encontrarla no requiere esfuerzo.
- Sugiere sin agredir: da más la sensación de buscar cooperación que de mostrar un afán dictatorial
- Es breve pero nunca ambigua.

Al igual que ocurre con los correos, la política debe escribirse en un lenguaje claro. El simple hecho de redactar una política no es suficiente. Instaurar una política no tiene sentido si no comprueba su eficacia. Debe comprobar de forma periódica lo siguiente:

Si los empleados siguen la política.
Si esta actualizada.
Si tiene suficiente alcance respecto a las mediadas de operación y seguridad.

Existen numerosas formas de comunicar y reforzar la política. Según el tamaño de la organización, puede seleccionar una, varias o incluso todas de las siguientes opciones.

- Un documento impreso
- Un documento hecho público en la red interna
- Una declaración firmada
- Notas que aparecen en pantalla

Revisar las políticas y los procedimientos existentes

Si ya se tiene una política, se deben revisar los siguientes puntos:

Los empleados puede acceder a la política fácilmente y en todo momento. Una política debe ser fácil de encontrar, de usar y de entender.

Se debe recordar a los empleados la existencia de la política con suficiente frecuencia.

El contenido de la política se revisa con suficiente frecuencia (cada seis meses) y se actualiza para reflejar cambios en: la tecnología, la legislación y los procedimientos de la organización. Es importante recordar que una política tiene escaso valor si no es eficaz y si no es capaz de evolucionar con los tiempos.

La organización revisa la política cada vez que ocurre un incidente no previsto en la misma.

La política de e-mail debe reducir la posibilidad de que se den los riesgos y de darse, ha de minimizar sus efectos.

4.3 Firmas digitales y Cifrado en los Mensajes

Para proteger la integridad y la confidencialidad de los mensajes de correo se puede usar la criptografía de diferentes maneras, por ejemplo:

- Firmar digitalmente un mensaje de correo para asegurar su integridad y confirmar la identidad del remitente.

- Encriptar el cuerpo de un mensaje para asegurar su confidencialidad.

- Encriptar la comunicación entre servidores de correo para proteger la confidencialidad tanto del cuerpo como de la cabecera del mensaje electrónico.

Los primeros dos métodos (la firma del mensaje y la encriptación del cuerpo) a menudo se usan de forma conjunta con el objetivo de que los destinatarios puedan confiar en la integridad del mensaje y a su vez verificar la identidad del remitente. El tercer método enlistado, sólo se usa cuando dos organizaciones quieren proteger los mensajes intercambiados entre ellos. Para ello se puede implementar una Red Privada Virtual (VPN) entre dos servidores de correo. Este método permite encriptar los elementos de la cabecera así como el cuerpo del mensaje. Sin embargo, una VPN no puede proteger un mensaje de correo a través de todas las rutas que puede tomar un mensaje electrónico. Es por ello que la mayoría de las veces se implementa la firma individual de los mensajes y, si se requiere, la encriptación de los mismos.

Los dos estándares más usados para firmar y encriptar el cuerpo de los mensajes son: OpenPGP (Open Pretty Good Privacy) y S/MIME (Secure/Multiporpose Internet Mail Extensions). Ambos métodos se basan (en parte) en la criptografía de llave pública lo que implica que un usuario debe tener un par de llaves. La primera llave se le llama pública y puede ser conocida por todos los usuarios, la segunda llave es privada lo que significa que sólo su propietario debe conocerla. También existe lo que se llama encriptado simétrico que consiste en tener una sólo llave que se comparte entre el remitente y el destinatario.

4.3.1 OpenPGP

OpenPGP es un protocolo para encriptar y firmar digitalmente mensajes de correo electrónico (también puede encriptar archivos). El funcionamiento de OpenPGP se basa en el protocolo PGP desarrollado por Phil Zimmerman.

Existen muchos productos libres y comerciales que usan OpenPGP, estos productos soportan una variedad de algoritmos de encriptación por

ejemplo utilizan: 3DES y AES para la encriptación de datos, DSA y RSA para las firmas digitales y SHA para crear cadenas de hash. Algunas implementaciones de OpenPGP soportan otros algoritmos de encriptación.

A pesar de que OpenPGP usa criptografía de llave pública para ciertos elementos como la firma digital, la encriptación del mensaje se lleva a cabo mediante un algoritmo simétrico. El procedimiento consiste en la generación de una llave aleatoria y la encriptación del mensaje con dicha llave mediante un algoritmo simétrico. El remitente cifra la llave simétrica con la correspondiente llave pública del destinatario. Una vez hecho lo anterior, se envía el mensaje y la llave simétrica, ambos cifrados, en el correo. Debido a que sólo el destinatario posee la llave privada (idealmente) nadie más estará en posibilidades de desencriptar el mensaje.

A continuación se enumeran los pasos que realiza OpenPGP para firmar y encriptar mensajes.

- OpenPGP comprime el texto del mensaje para reducir el tiempo en la transmisión y fortalecer la seguridad.
- OpenPGP crea una llave aleatoria de sesión.
- Se crea una firma digital para el mensaje a partir de la llave privada del remitente y se agrega al mensaje.
- El mensaje y la firma son encriptadas usando la llave de sesión y un algoritmo simétrico (3DES, AES)
- La llave de sesión es encriptada usando la llave pública del destinatario y es anexada al inicio del mensaje encriptado.
- El mensaje encriptado es enviado al destinatario.

En el otro extremo de la comunicación se realiza el proceso inverso. Se recupera la llave de sesión a través de la llave privada, se desencripta el mensaje y se verifica la firma digital. Muchos de los agentes de usuario de correo soportan el encriptado de mensajes con OpenPGP, aunque algunos de ellos necesitan “plug-ins” para que funcione de manera adecuada.

4.3.2 S/MIME

S/MIME es un protocolo para cifrar y firmar los mensajes de manera digital. Está basado en el formato PKCS para el cifrado de datos y en X.509 para la creación de certificados digitales. En sus inicios, el protocolo S/MIME fue propuesto por RSA Data Security Inc.
S/MIME fue creado basándose en el protocolo existente MIME y puede ser integrado fácilmente en los productos existentes del e-mail y de la mensajería.

El proceso de encriptación de S/MIME es muy similar al que usa OpenPGP. La versión 3.1 de S/MIME soporta dos algoritmos simétricos de cifrado, AES y 3DES.

4.3.3 ADMINISTRACIÓN DE LLAVES

OpenPGP y S/MIME usan certificados digitales para el manejo de las llaves. Un certificado digital identifica entidades (usuario, organizaciones). El certificado digital contiene usualmente el nombre de la entidad certificada, un número serial, fecha de expiración, una copia de la clave pública del titular del certificado (utilizada para la verificación de su firma digital), y la firma digital de la autoridad emisora del certificado de forma que el receptor pueda verificar que esta última ha establecido realmente la asociación. A pesar de que ambos protocolos utilizan certificados digitales la diferencia entre ellos radica en el modelo de administración de llaves que utiliza cada uno para establecer el uso de certificados digitales confiables.

El modelo que usa OpenPGP para el manejo de sus llaves se le conoce como “red de confianza” el cual se basa en niveles de confianza individuales que se van extendiendo de usuario a usuario. Por ejemplo, si José confía en Juan y Luis confía en José, entonces Luis debe confiar en los mensajes de Juan. Lo anterior es adecuado para usuarios individuales y para organizaciones pequeñas, la desventaja del proceso es que muchas veces no es funcional para organizaciones muy grandes. Algunas organizaciones mantienen servidores de llaves para que los usuarios puedan obtener llaves públicas y almacenar las propias. Este método permite centralización, sin embargo, el proceso está controlado principalmente por usuarios individuales y las organizaciones no se sienten cómodas confiando en servidores de llaves para certificar las identidades de los usuarios.

Por otro lado, S/MIME maneja las llaves a través del modelo de confianza en autoridades certificadoras (CA). En este modelo existe un registro a una autoridad aprobada la cual se le conoce como Autoridad certificadora raíz, esta autoridad garantiza la vinculación entre la identidad de un sujeto o entidad y su clave pública.

S/MIME y OpenPGP son métodos confiables y probados. La selección de uno u otro depende del grado de confianza de una organización en sus usuarios o en una entidad certificadora.

Siempre se debe recordar que, aunque tenga garantías de la confidencialidad del mensaje y de la identidad del emisor, el e-mail puede contener virus. Es posible que ni el emisor ni el destinatario lo averigüen hasta que sea demasiado tarde.

4.3.4 DESVENTAJAS EN EL CIFRADO DE CORREO ELECTRÓNICO

A pesar que el cifrado del correo electrónico proporciona seguridad, también tiene ciertas desventajas. Las organizaciones deben tener en cuenta las siguientes desventajas en la implementación del cifrado de mensajes:

- El cifrado y descifrado de mensajes requieren tiempo de procesador. Las organizaciones necesitan equipos capaces de soportar esta carga extra.

- El uso del cifrado requiere administración adicional para la distribución, la recuperación y la revocación de llaves cifradas.

- El cifrado de correo complica la revisión de los datos que viajan en él.

- Algunos mensajes cifrados, enviados o recibidos, pueden estar mal protegidos si la organización no implementa el uso de algoritmos de encriptación fuertes o el tamaño de las llaves no es el adecuado, con lo cual la información enviada no es tan segura como aparenta.

Cuando se usa cifrado de mensajes se complica la búsqueda de malware tanto en el firewall como en el servidor de correo. Si el firewall o el servidor de correo no tienen un método para desencriptar los mensajes, no podrán ser capaces de actuar contra virus u otros agentes maliciosos anexos al correo. Algunos programas que revisan malware pueden ser habilitados para trabajar en el momento que se desencriptan los

correos, sin embargo, esta solución es compleja y trae consigo la probabilidad de que el programa quede infectado por algún agente malicioso. Por lo tanto, si no es posible tener algún programa para desencriptar los correos y revisarlos en busca de malware (en el servidor), esta operación debe quedar en manos de los programas clientes de correo.

4.4 PLANEACIÓN Y MANEJO DE LA SEGURIDAD EN LOS SERVIDORES DE CORREO

El aspecto más crítico en la implementación de un servidor de correo electrónico es la planeación antes de la instalación y configuración de este servicio. Una cuidadosa planeación en la implementación de un servicio debe contemplar, entre otras cosas, un análisis de la seguridad y el cumplimiento de las políticas de la organización. Muchos problemas de rendimiento y seguridad en un servidor de correo son consecuencia de la falta de planeación y de control en la administración. Por lo tanto, la seguridad del servidor debe considerarse desde la fase de planeación ya que es mucho más difícil implementarla en un servidor en operación. Además, cuando se tiene un plan definido y bien detallado es fácil tomar decisiones acerca de la utilidad, rendimiento y riegos que conllevan la instalación de cualquier servicio.

En la fase de planeación de un servidor de correo electrónico se deben considerar los siguientes puntos:

- Identificar el propósito del servidor de Correo.
  ¿Qué tipo de información será almacenada, procesada y transmitida a través del servidor de correo?
  ¿Cuáles son los requerimientos de seguridad para esa información?
  ¿Qué otros servicios serán proporcionados por el servidor de correo (en general, un equipo dedicado para el correo es la opción más segura)?
  ¿Cuáles son los requerimientos de seguridad para los servicios adicionales?

  ¿Cuáles son los requerimientos para la disponibilidad del servicio?
  ¿En qué parte de la red estará colocado el Servidor de Correo?

- Identificar los servicio de red adicionales que tengan relación con el correo (POP, IMAP, Web-Mail, etc.)

- Identificar los usuarios o categorías de usuarios que existirán en el servidor de correo.

- Determinar los privilegios que cada categoría de usuarios tendrá en el servidor de correo.

- Determinar la forma en que el servidor de correo será administrado (local, remotamente dentro de la red, remotamente de forma externa).

- Determinar la manera en que los usuarios serán autenticados y cómo proteger los datos de la autenticación.

- Decidir qué tipo de información será asegurada, por ejemplo, cuentas de usuarios, identidad de los usuarios, así como la relación de ciertos usuarios con la organización.

- Considerar que implantar medidas severas de seguridad puede reducir la funcionalidad del servidor.

Otros aspectos a considerar en la planeación son:

Costos.
Compatibilidad con la infraestructura existente.
Habilidades de los administradores.
Historial de Vulnerabilidades.
Funcionalidad.

La elección del software que soportará al servidor de correo muchas veces determina la elección del sistema operativo, sin embargo, se debe buscar un sistema operativo que proporcione:

- Rápida solución de vulnerabilidades.

- Capacidad para restringir actividades administrativas sólo a los usuarios autorizados.

- Capacidad para proteger información privada en el servidor.

- Facilidad para deshabilitar servicios de red innecesarios que vienen por default durante la instalación de software.

- Capacidad para registrar la actividad del servidor y con ello detectar intrusiones o intentos de intrusión.

Aparte de los puntos comentados, las organizaciones deben considerar la existencia de personal con experiencia y capacidad para administrar el servidor así como los productos relacionados con él.

Dada la naturaleza crítica del servicio de correo es importante que el equipo destinado para ello se encuentre localizado en un área que proporcione un ambiente físicamente seguro. Cuando se planea la localización del servidor se deben contestar las siguientes preguntas:

- ¿La localización propuesta ofrece mecanismos de protección física adecuados? Algunos ejemplos incluyen lectores de tarjetas, guardias de seguridad y sistemas de detección de intrusiones (cámaras, sensores de movimiento, etc.)
- ¿El lugar tiene controles ambientales para mantener la temperatura y la humedad convenientes para el servidor?
- ¿Existen fuentes de energía de respaldo (no-break)?
- ¿El sitio es el idóneo en caso de algún desastre natural?

4.4.1 RESPONSABILIDADES DE SEGURIDAD DEL ADMINISTRADOR

El administrador de correo es responsable de las siguientes actividades de seguridad relacionadas con el servidor.

- Instalar y configurar el equipo cumpliendo con las políticas de seguridad impuestas por la organización y los procedimientos estándar recomendado por expertos.

- Mantener el servidor con procedimientos seguros, incluyendo respaldos frecuentes, aplicación de parches y actualizaciones

- Monitorear la integridad del sistema, los niveles de protección y eventos relacionados con seguridad.

- Revisar el sistema en busca de anomalías asociadas con los recursos del sistema.

- Realizar pruebas de seguridad de forma periódica.

4.4.2 PROCEDIMIENTOS DE ADMINISTRACIÓN

Los procedimientos de administración del servidor son esenciales para la operación y mantenimiento de la seguridad del servicio. Las prácticas de seguridad involucran la identificación de los sistemas de información de la organización y el desarrollo, documentación, y la implementación de políticas, procedimientos, y guías para lograr la confidencialidad, integridad, y disponibilidad de dichos sistemas de información.

Cuando una organización pretende brindar seguridad a un servidor de correo debe implementar los siguientes puntos:

Política de Seguridad de los Sistemas de Información—Es el conjunto de reglas de lo que se puede y no se puede hacer en el área de seguridad durante la operación de un sistema. La política de seguridad indica los responsables de la seguridad de la información. El documento que contiene las políticas de seguridad debe ser actualizado constantemente.

Control y Cambios de Configuración—El control en las configuraciones del sistema durante el diseño previene contra modificaciones erróneas durante la instalación y operación del servidor, ya que ciertas modificaciones pueden introducir riegos de seguridad.

Evaluación y Administración de Riesgos—Evaluación de riesgos es el proceso de analizar e interpretar los riesgos. Para realizar una evaluación de riesgos es necesario recolectar datos relacionados con los riesgos, por ejemplo, vulnerabilidades, fortalezas y las consecuencias de un ataque exitoso. La administración de riesgos es el proceso de seleccionar e implementar mecanismos de control para reducir los riesgos a un nivel aceptable por la organización.

Recomendaciones de Seguridad—Las organizaciones deben desarrollar un conjunto de recomendaciones (comprobadas) para asegurar los datos de un sistema. Estas recomendaciones proporcionan una guía para el administrador del sistema en la forma de configurar el equipo para cumplir con los mecanismos de seguridad impuestos por las políticas de la organización.

Entrenamiento en Seguridad—Un programa de capacitación en seguridad es importante para proporcionar los elementos técnicos acerca de la implementación de la seguridad en los diferentes sistemas de información. Además, un entrenamiento de seguridad hace

conscientes a los administradores acerca de las responsabilidades de seguridad que cada uno tiene. Con ello, se cambian comportamientos y procedimientos en la operación de un sistema con la finalidad de reforzar la seguridad.

Plan de Contingencia y Recuperación de Desastres—Un plan de contingencia y de recuperación de desastres se establecen con la finalidad de mantener el funcionamiento del servicio o la rápida recuperación del mismo en caso de alguna eventualidad.

Plan de seguridad del sistema

El objetivo de un plan de seguridad es proporcionar protección a un sistema delineando responsabilidades, estableciendo mecanismos de control de seguridad y describiendo el comportamiento esperado de todos los individuos involucrados en el sistema de correo. El plan se realiza para proteger de manera adecuada la información y los recursos del sistema. Es recomendable que todos los sistemas tengan un plan de seguridad.
En general, un plan de seguridad efectivo debe incluir lo siguiente:

- **Descripción del Sistema.** La primera sección de un plan de seguridad proporciona una básica descripción acerca del sistema. Esta primera sección debe contener información general del sistema, por ejemplo, el propósito del sistema, el nivel de sensibilidad de la información que maneja, los puntos de entrada al sistema y el ambiente en el cual funcionará.

- **Controles.** Esta sección del plan describe las medidas de control para el sistema.

Los mecanismos de control caen en tres categorías.

- **Administración de los mecanismos de control**. Se enfoca en los procedimientos utilizados para asegurar un sistema y en el manejo de riesgos del sistema.

- **Controles de Operación.** Este tipo de controles los llevan a cabo los administradores, para ello requieren un alto conocimiento de la seguridad en general y del sistema.

- **Controles técnicos.** Son los mecanismos de seguridad que el sistema emplea. Es decir, las herramientas automatizadas para

proporcionar protección de accesos no autorizados. Este tipo de controles facilitan la detección de una violación del sistema, de un mal uso de la información, etc.

4.4.3 PRINCIPIOS GENERALES DE SEGURIDAD EN LOS SISTEMAS DE INFORMACIÓN

Cuando se piensa en implantar mecanismos de seguridad en cualquier sistema de información electrónica, se deben tener en cuenta algunos principios básicos como son los siguientes:

Simplicidad—Los mecanismos de seguridad deben ser lo más simple posible. La complejidad es la raíz de muchos problemas de seguridad.

Fallos de Forma Segura—Si ocurre un fallo, el sistema sólo debe fallar en funcionalidad pero no en la parte de seguridad.

Acceso Indirecto—En lugar de proporcionar acceso directo a la información, se deben implementar, en la medida de lo posible, mecanismos intermedios como lo son firewalls, proxy's, permisos del sistema, etc.

División de Privilegios—El concepto de división de privilegios se debe aplicar tanto a las funciones del sistema como a los usuarios encargados de ejecutar ciertas tareas. En el caso de los sistemas, las funciones como leer, editar, escribir y ejecutar deben tener distintos permisos. En el caso de los usuarios se deben asignar roles dependiendo de las tareas a ejecutar.

Mínimo Privilegio—Este principio determina que cada proceso o usuario debe tener el permiso mínimo requerido para realizar su tarea. Una aplicación adecuada de este principio evita que el daño de un proceso o cuenta comprometida se propague más allá de los recursos disponibles para esa entidad.

Aceptación de Mecanismos de Seguridad—Los usuarios del sistema deben comprender la necesidad de la implantación de seguridad. En ocasiones un mecanismo de seguridad minimiza la flexibilidad del sistema. En este caso se deben proporcionar a los usuarios opciones alternas para seguir disfrutando de cierta funcionalidad.

Seguridad en Capas—Las organizaciones deben implementar varios mecanismos de seguridad para que la violación de alguno de ellos no sea suficiente para comprometer el servidor en su totalidad.

Registro de Ataques—Se debe mantener un registro de los ataques ocurridos a un sistema. Esta información puede ayudar a identificar los métodos usados por los atacantes, para que en un futuro no se vuelvan a repetir. Con ello se puede reforzar la seguridad de sistemas próximos a implementarse.

4.5 SEGURIDAD EN EL SISTEMA OPERATIVO

Proteger un servidor de correo implica la protección del sistema operativo que lo soporta. Muchos inconvenientes relacionados con la seguridad de un servicio se podrían evitar si se instala y se configura de manera adecuada el sistema operativo en el cual está instalado dicho servicio.

La mayoría de los sistemas operativos vienen configurados de manera predeterminada para resaltar las características, funcionalidad y facilidad de uso, todo ello a costa de la seguridad. Debido a lo anterior, es responsabilidad de los administradores de sistemas la configuración de los servidores para satisfacer los requerimientos de seguridad impuestos por una organización.

Obviamente las configuraciones de seguridad varían de acuerdo al sistema utilizado, incluso existen herramientas automatizadas para fortalecer la seguridad de ciertos sistemas operativos. Sin embargo, cinco pasos son necesarios para mantener la seguridad básica de cualquier sistema operativo.

1. Una planeación de la instalación del sistema operativo así como del software adicional para proporcionar el servicio requerido.

2. Actualizar el sistema operativo según se requiera.

3. Configuración del sistema operativo para cumplir las políticas de seguridad.

4. Instalar y configurar mecanismos de seguridad adicionales en caso de ser necesario.

5. Realizar pruebas que determinen el cumplimiento de los cuatro pasos anteriores.

4.5.1 ACTUALIZACIÓN E INSTALACIÓN DE PARCHES PARA EL SISTEMA OPERATIVO

Una vez que el sistema operativo está instalado es necesario aplicar los parches correspondientes y hacer las actualizaciones que recomienden los desarrolladores del software. Cualquier vulnerabilidad conocida debe ser corregida antes de liberar el equipo como servidor de correo, por lo tanto, los administradores deben:

- Crear e implementar un procedimiento de instalación de parches.
- Identificar vulnerabilidades y aplicar los correspondientes parches
- Reducir de forma temporal las vulnerabilidades hasta que existan parches que las corrijan.

Los administradores deben asegurarse que los servidores recién instalados deben de ser protegidos durante el proceso de actualización. Un servidor que no está totalmente actualizado puede ser comprometido si es accesible públicamente durante el proceso de configuración. Por lo anterior, cuando se prepara la instalación de un servidor se debe mantener el equipo desconectado de la red o conectado dentro de un segmento seguro (VLAN) hasta que el proceso de actualización y configuración haya finalizado.

Los administradores no deben aplicar parches sin haberlos probado sobre otro servidor idéntico debido a que ciertos parches pueden ocasionar problemas en la operación del servicio. A pesar de que los administradores pueden configurar un servidor de correo para descargar los parches de manera automática no es recomendable configurar el servidor para instalar parches de manera automática sin probarlos de manera previa.

4.5.2 REMOVER O DESHABILITAR APLICACIONES Y SERVICIOS INNECESARIOS

Idealmente, un servidor de correo electrónico debe ser un equipo dedicado a ese servicio. Por lo tanto, cuando se configura el sistema operativo se debe deshabilitar todos los servicios excepto aquellos que están expresamente permitidos. Lo anterior significa que, es necesario deshabilitar todos los servicios y aplicaciones que no tengan relación con el servicio de correo electrónico o que no proporcionen soporte alguno a ese servicio.

Algunos servicios y aplicaciones que usualmente se deshabilitan porque no se requieren en el servidor de correo son:

- Servicios de intercambio de archivos y de impresión, por ejemplo, FTP, NFS, CUPS, etc.
- Servidores Web instalados por default.
- Soporte para redes inalámbricas (en caso de no tener hardware para ello).
- Programas para controlar y acceder remotamente al sistema, especialmente aquellos que no encriptan de manera adecuada la comunicación.
- Servicios de directorio innecesarios (NIS, NIS+, Kerberos, etc.)
- Compiladores y librerías no necesarios.
- Herramientas de desarrollo del sistema.
- Herramientas y utilidades para la administración de la red (SNMP).

Desinstalar servicios y aplicaciones innecesarias es preferible que deshabilitarlas debido a que ciertos ataques informáticos alteran aplicaciones o activan servicios, lo que no ocurriría si los componentes de las aplicaciones fueran completamente removidos. Además, un error humano puede habilitar un servicio innecesario.

Cada servicio que se agrega al servidor de correo incrementa el riesgo de seguridad, porque abre un nuevo canal que puede ser aprovechado por un atacante si la configuración no es la correcta. Todos los servicios adicionales incrementan la carga del sistema lo que puede afectar el rendimiento del mismo.

Cuando se reducen los servicios en un servidor de correo también se reducen los registros de la actividad del sistema lo que facilita encontrar problemas en el funcionamiento del servicio.

Por lo anterior, las organizaciones deben determinar los servicios que serán habilitados en un servidor de correo. Algunos servicios pueden requerirse en ciertas circunstancias (administración remota, acceso a directorios, correo vía Web, etc.). Estos servicios deben ser cuidadosamente configurados para evitar problemas en la seguridad de todo el equipo.

4.5.3 AUTENTICACIÓN DE USUARIOS

Habilitar la autenticación de usuarios en el sistema involucra la configuración de ciertos elementos del sistema operativo. Para evitar problemas en la seguridad de las cuentas de usuarios que tienen acceso al sistema se pueden usar mecanismos de autenticación a nivel hardware o software. Lo que se recomienda es que siempre exista un método de autenticación y que los datos intercambiados entre el servidor y el usuario viajen cifrados.

Para implantar cierto grado de seguridad en la autenticación de usuarios, los administradores deben llevar a cabo ciertas tareas las cuales incluyen las siguientes:

Remover o deshabilitar cuentas de usuarios instaladas por default. Cuando se instala un sistema operativo se habilitan cuentas por default, estas cuentas pueden ser de usuarios invitados, de administradores o de servicios (locales y de red). Las cuentas por default se deben borrar del sistema. En caso de que dichas cuentas sean indispensables para el funcionamiento del equipo se les debe asignar contraseñas adecuadas.

Deshabilitar cuentas no interactivas. Existen cuentas que son indispensables para el correcto funcionamiento del sistema pero que no necesitan un intérprete de comandos. A este tipo de cuentas se les debe impedir la comunicación interactiva con el sistema operativo.

Crear grupos. En un sistema donde existen muchos usuarios, se deben crear grupos y asignar los usuarios al grupo de acuerdo al rol que jueguen dentro de la organización, para así, asignarles permisos de acuerdo a su jerarquía.

Crear cuentas de usuarios. Cada usuario que pueda gozar del servicio que proporciona el servidor debe tener una cuenta propia, no es recomendable que los usuarios compartan cuentas porque pueden realizar operaciones a nombre del usuario con quien comparte la cuenta.

Revisar la política de contraseñas. Una contraseña adecuada dificulta a un atacante comprometer las cuentas, por ello es importante contemplar los siguientes aspectos en la configuración de contraseñas.

- **Tamaño** – Las contraseñas deben tener un tamaño mínimo. Ocho caracteres es considerado un tamaño mínimo adecuado.

- **Complejidad** – Las contraseñas deben estar constituidas por la combinación de letras (mayúsculas y minúsculas) números y caracteres no alfanuméricos, además se debe vigilar que la contraseña no sea una palabra de diccionario.

- **Tiempo de vigencia** –Los usuarios deben cambiar sus contraseñas de forma periódica. El tiempo de vigencia indica la frecuencia con que debe cambiarse la contraseña.

- **Reutilización** – Es importante recomendar a los usuarios que no reutilicen las contraseñas, muchos usuarios cambian las contraseñas simplemente agregando caracteres al final de su contraseña actual. Lo anterior no es una práctica muy segura y debe evitarse.

Evitar intentos automáticos de penetración. Es relativamente fácil para un usuario sin autorización tratar de introducirse al sistema usando herramientas automáticas que intentan adivinar contraseñas. Con lo anterior, los administradores deben configurar el sistema para incrementar el tiempo entre intentos de acceso. Otra posible solución es deshabilitar una cuenta si ocurre un número determinado de intentos de acceso fallidos.

Instalar y configurar otros mecanismos que refuercen la autenticación. En caso que la información del servidor de correo lo requiera es prudente considerar otros métodos de autenticación, por ejemplo, métodos biométricos, tarjetas inteligentes, tokens, etc. Dichos métodos pueden ser caros y difíciles de implementar pero pueden estar justificados en ciertas circunstancias. Cuando se implementan mecanismos de autenticación extra, se debe cambiar la política de acuerdo a la nueva implementación.

Muchos atacantes usan herramientas para capturar contraseñas mientras éstas viajan por la red, un método para evitar ser víctimas de robo de contraseñas es usar tecnologías de encriptación de datos mientras viajan por la red (SSL, TLS, SSH, VPN, etc.).

4.5.4 CONFIGURACIÓN DEL CONTROL DE RECURSOS DEL SISTEMA

Todos los sistemas operativos modernos pueden limitar los privilegios de sus elementos, por ejemplo es posible limitar el acceso a archivos, directorios, dispositivos y recursos de la máquina en general. Lo anterior evita que usuarios sin privilegios puedan acceder a la información

privada dentro del sistema y ejecutar procesos que pueden provocar una denegación del servicio que ofrece el equipo.

4.5.5 INSTALAR Y CONFIGURAR ELEMENTOS ADICIONALES DE SEGURIDAD

A pesar de que los sistemas operativos incluyen software de seguridad, en ocasiones será necesario seleccionar, instalar y configurar elementos adicionales que ayuden a reforzar la seguridad del sistema. El software adicional a instalar debe incluir: anti-malware, detectores de intrusos de host, firewall de host y herramientas de análisis de vulnerabilidades.

Algunos administradores de correo instalan herramientas de detección de intrusos de host, por ejemplo, software que detecta la integridad de los archivos críticos así como sus modificaciones. Antes de instalar herramientas adicionales de seguridad, se debe evaluar el impacto que tendrán en el rendimiento del sistema.

4.6 SEGURIDAD EN EL SERVICIO DE CORREO ELECTRÓNICO

Para brindar un grado de seguridad aceptable al servidor de correo es fundamental proteger el contenido de los mensajes que viajan a través del medio de comunicación. La protección de contenido incluye el filtrado, búsqueda de malware, combate al spam (correo no deseado). También es importante proteger los buzones de coreo de los usuarios e implantar mecanismos de seguridad para el acceso a mensajes a través de Web. La seguridad en el contenido de mensajes también incluye el cifrado para mantener la confidencialidad y las firmas digitales para soportar la integridad y el no repudio.

Después de asegurar el sistema operativo del equipo, el siguiente paso es instalar el software que proporcionará el servicio de correo, y a su vez, configurarle aplicaciones de seguridad.

4.6 INSTALACIÓN DEL SERVICIO DE CORREO DE FORMA SEGURA

Una instalación y configuración segura de la aplicación del servidor de correo es muy parecida al proceso que se sigue para asegurar el sistema operativo. A grandes rasgos los puntos fundamentales para asegurar la aplicación del servidor de correo son los siguientes:

- Instalar el software del servidor de correo sobre un equipo dedicado.

- Aplicar parches o actualizaciones para corregir vulnerabilidades conocidas.

- Crear un disco físico dedicado o una partición lógica (independiente de la aplicación del servidor de correo) para buzones de correo, o mejor aún, instalar los buzones en otro servidor.

- Desinstalar o deshabilitar todos los servicios innecesarios instalados por el software que proporciona el servicio de correo.

- Remover o deshabilitar todas las cuentas de usuarios creadas por la instalación del servicio de correo y que nos son necesarias en su operación.

- Reubicar o desinstalar toda la documentación relacionada con el software del servidor de correo.

- Reubicar o borrar cualquier archivo de ejemplo relacionado con el funcionamiento del servidor de correo.

- Instalar cualquier script recomendado por el desarrollador del software para reforzar la seguridad.

- Cambiar la información mostrada por los servicios de red del servidor, debe evitarse reportar el tipo y versión tanto del software como del sistema operativo (en la medida de lo posible).

- Deshabilitar comandos innecesarios y peligrosos para la seguridad (VRFY, EXPN)

- Cambiar la configuración por default de los servicios instalados para la operación del correo.

4.6.2 CONTROL DE ACCESO AL SERVICIO DE CORREO

Los controles de acceso adecuados pueden prevenir la exposición de información sensible y privada en el servidor de correo. Además, una buena configuración en los recursos del sistema ayuda a evitar ataques de denegación de servicio contra el servidor de correo.

Los siguientes archivos deben tener un control estricto de acceso.

- El software de la aplicación y sus archivos de configuración.

- Directorios relacionados de forma directa con la seguridad:
  Archivos de contraseñas y otros archivos relacionados con la autenticación.
  Archivos que contienen información usada en el control de acceso.
  Llaves usadas en el cifrado y en las firmas digitales.

- Registros del servidor y archivos útiles en auditoria informática.

- Software del sistema y sus archivos de configuración.

La aplicación que proporciona el servicio de correo se debe ejecutar bajo una cuenta de usuario y un grupo con restricciones estrictas de acceso. Por lo tanto, se debe crear una nueva cuenta y un nuevo grupo para asignarlos de manera exclusiva al software de correo. El nuevo usuario y grupo deben ser independientes y únicos. Otra situación que se tiene que vigilar es el acceso del software de correo a los archivos del sistema. En caso de que sea posible, los procesos del servidor de correo deben tener acceso de solo lectura a los archivos que ayuden en su funcionamiento, y no tener ningún tipo de acceso a otros archivos del servidor, por ejemplo, los archivos de registro del sistema.

Algunas sencillas configuraciones para limitar el uso de recursos del sistema operativo que puede usar el software de correo ayudan a evitar ataques de denegación de servicio.
Algunos ejemplos de limitaciones útiles, son:

- Instalar los buzones de usuario en un servidor, disco duro, o partición lógica diferente de donde está instalada la aplicación del sistema de correo.
- Configurar la aplicación del servidor de correo para que no pueda consumir todo el espacio de disco o partición donde está instalado.

- Limitar el tamaño de archivos adjuntos que se pueden mandar en un mensaje

- Vigilar que los archivos de registro están instalados en un lugar con suficiente espacio para contenerlos.

4.6.3 PROTECCIÓN DEL CORREO CONTRA AGENTES MALICIOSOS (MALWARE)

Una de las mayores ventajas del correo electrónico es que permite enviar archivos binarios junto con el mensaje. Lo anterior incrementa la productividad en una organización ya que se pueden mandar hojas de cálculo, presentaciones, reportes, diagramas imágenes, etc. Sin embargo, muchas formas de agentes maliciosos (virus, gusanos, caballos de troya, spyware) también pueden ser enviados junto al mensaje. Este tipo de agentes buscan perjudicar la productividad del sistema, ganar acceso privilegiado, robar información, instalar aplicaciones, entre otras cosas. Por lo tanto, para proteger el correo contra agentes maliciosos es fundamental instalar programas que puedan revisar los mensajes en busca de malware, también es útil aplicar un filtrado de contenido y crear procedimientos que ayuden a los usuarios a identificar contenido dañino.

Revisión de mensajes en busca de malware

Para protegerse contra virus, gusanos y otras formas de malware, es importante realizar revisiones de mensajes en algunos puntos durante el proceso de entrega del correo. La revisión de malware se puede implementar en el firewall, en el servidor de correo y/o en los equipos clientes del correo. Como mínimo, se deben implementar revisiones en el servidor de correo y en las máquinas clientes.

Revisión en el servidor de correo

La revisión de los mensajes en el servidor de correo proporciona ciertas ventajas:

- Se pueden revisar los mensajes entrantes y salientes (ambas direcciones)
- Las revisiones se hacen de forma centralizada lo que facilita la actualización del software.
- Protege tanto al servidor como a los clientes de correo.

Las desventajas de implementar revisiones en el servidor son:

- Se necesita configurar el software que actúa como servidor de correo para que conviva con el escáner de mensajes.
- No se pueden revisar mensajes cifrados.

- Existe una carga extra en el sistema que puede afectar el rendimiento del mismo.

El software encargado de revisar los mensajes en busca de malware debe tener las siguientes características:

- Detector y limpiar todo tipo de malware.
- Proporcionar algún tipo de protección contra malware nuevo y desconocido
- Proporcionar filtrado de contenido.
- Facilidad de Uso.
- Facilidad de descarga e instalación automática de actualizaciones.
- Proporciona actualizaciones frecuentes
- Proporciona mecanismos robustos de alertas.
- Registra toda su actividad.

Revisión en las máquinas cliente

El escáner de malware puede ser instalado en los dispositivos clientes de correo. De esta forma, los correos entrantes serán revisados cuando los usuarios los intenten abrir, y los correos salientes serán revisados cuando los usuarios los intenten enviar. La principal ventaja de este tipo de configuración es que la revisión de los mensajes se distribuye en muchos equipos con lo que no existe un impacto importante en el rendimiento del dispositivo.

4.6.4 FILTRADO EN EL CONTENIDO DE LOS MENSAJES

El filtrado de contenido busca mensajes que contienen elementos de malware (spam, hoaxes, lenguaje ofensivo o inapropiado para la organización, etc.). El filtrado de contenido se puede realizar en los mismos puntos en donde se realiza la revisión de malware. De hecho, existe software que puede realizar ambas operaciones (filtrado de contenido y revisión de malware).

De forma general, un filtrado de contenido se lleva a cabo mediante un análisis léxico de los mensajes y se compara contra palabras o frases definidas en reglas que establece la organización. Dichas reglas pueden estar relacionadas con contenido ofensivo o con información privada de la organización. El destino de un mensaje que coincida con las reglas impuestas en el filtrado depende de la configuración impuesta por el administrador, los mensajes pueden ser aislados, borrados, regresados, etc.

Prácticas para evitar contenido malicioso

Adicionalmente a la implantación de anti-malware, los usuarios deben seguir ciertos procedimientos para evitar que sus equipos se contaminen por un agente malicioso proveniente de algún mensaje electrónico.
Los procedimientos a seguir se enlistan a continuación:

- Nunca abrir mensajes adjuntos enviados por desconocidos.

- Nunca abrir mensajes adjuntos con contenido potencialmente peligroso (archivos con extensión .exe, vb)
- No abrir archivos comprimidos y con contraseña, las cuales vienen en el texto del mensaje. Se está volviendo cotidiana esta técnica con la cual las firmas antivirus son evadidas.
- No se deben abrir correos con ligas a supuestas tarjetas virtuales o videos animados, debido a que esto podría ser una forma de introducir un virus al equipo y comprometer la seguridad del mismo.
- No abrir mensajes que en el apartado de asunto tenga frases sospechosas.
- Revisar todos los mensajes de correo con algún anti-malware antes de abrirlos.
- Actualizar la base de datos (periódicamente) del software que se encarga de revisar el malware
- Ignorar mensajes que solicitan información personal o financiera.
- Romper cadenas para evitar que se propague el Spam.

4.6.5 COMBATE AL SPAM

En casi todos los medios de comunicación existen entidades tratando de expresar sus ideas o hacer publicidad de sus productos, el correo electrónico no es la excepción. El término usado para ese tipo de mensajes publicitarios es: correo comercial no solicitado, mejor conocido como “spam”.

Todos los días muchos usuarios del correo electrónico a lo largo del mundo reciben varios mensajes que contienen spam, incluso se dice que más de la mitad del coreo electrónico que circula por la red, es spam.

El correo spam es molesto para todas las organizaciones ya que hace perder la productividad de los empleados al leer correos que no aportan valor a la organización y en ciertos casos, dichos correos pueden saturar los buzones de los usuarios e incluso saturar el espacio en el servidor de

correo. Por lo tanto, las organizaciones buscan implantar medidas para reducir la cantidad de spam que los usuarios reciben diariamente.

Algunas medidas que pueden tomar los usuarios para reducir el spam son:

- Implantar filtros Anti-Spam (filtrado de contenido) en sus programas gestores de correo.
- Romper Cadenas de Correo.
- No proporcionar la dirección de correo en páginas no confiables.
- Nunca responder a un mensaje Spam.
- Tratar de enviar copias de correo de forma que se oculten las direcciones.

Los administradores de correo pueden adoptar las siguientes medidas contra el spam.

- Implementar filtros de correo para mensajes entrantes y salientes.
- Bloquear direcciones que reporten los usuarios como fuentes de Spam (después de comprobarlo).
- Bloquear direcciones de servidores conocidos por enviar spam.
- No exponer cuentas de usuarios en sitios públicos.
- Actualizar las listas negras.

Algunas organizaciones y sitios en Internet han creado listas de servidores conocidos por enviar mensajes publicitarios no solicitados. Dichas listas se actualizan diariamente y se les conoce como listas negras. Una forma de reducir en gran medida el spam es implantando un filtrado de direcciones por medio de listas negras.

4.6.6 REENVÍO DE MENSAJES CON AUTENTICACIÓN (RELAY)

Cuando un servidor de correo se usa para mandar mensajes desde cualquier dirección hacia cualquier dominio se le conoce como servidor de relay abierto. El relay no es más que un reenvío de mensajes a través de un servidor de correo. El relay puede ser útil si se configura de forma adecuada, por el contrario, si el servidor se configura de manera errónea con respecto al relay, puede ser un medio para enviar Spam. Desafortunadamente los servidores de relay abierto se utilizan por terceros para enviar Spam. Para solucionar esto, se crearon listas negras en tiempo real que bloquean hosts en los cuales se detecta un MTA que hace “Open Relay**”**.

Con lo anterior, se deben emplear métodos para permitir el relay solamente a los usuarios autorizados.

El primer método consiste en controlar el relay a través de subredes o dominios desde donde los usuarios autorizados envían correo. Sin embargo, si los usuarios se conectan de distintos dominios el método no es útil.

El segundo método requiere que los usuarios se autentiquen antes de enviar mensajes, a este método se le conoce como relay autenticado o SMTP AUTH. Para implantar este último método se deben hacer configuraciones extras al servidor de correo.

4.7 ADMINISTRACIÓN DE LA SEGURIDAD EN EL SERVIDOR DE CORREO

Después de instalar un servidor de correo, los administradores necesitan administrar la seguridad del sistema en forma continua. Las acciones más importantes que debe contemplar un administrador para mantener seguro un sistema son: Análisis de registros del sistema, respaldos periódicos de la información del sistema, recuperación en caso de eventos de seguridad y pruebas de seguridad del servidor. También es necesario asegurar la administración remota del sistema si es que se necesita gestionar el sistema de forma no presencial.

4.7.1 REGISTROS

Capturar los datos de la actividad del sistema y monitorearlos es vital para la correcta administración de la seguridad y rendimiento de un equipo. Habilitar mecanismos de registro ayuda a encontrar, entre otras cosas, fallos en el sistema, intentos de intrusión o ataques exitosos.

En el tema de la seguridad, un análisis de registros puede ser útil en ciertos casos.

- Encontrar actividades sospechosas que requieren de investigación detallada.
- Rastreo de las tareas realizadas por atacantes.
- Ayuda en la recuperación del sistema.
- Ayuda en la investigación después de una intrusión.

Registros del servidor de correo

Cada software de correo tiene capacidades diferentes en cuanto al tipo y detalle de la información que registra. Lo recomendable es configurar el nivel de registro para que al menos contenga lo siguientes elementos:

Registros relacionados con el sistema
- Problemas en la configuración de red.
- Errores en la configuración del servidor de correo.
- Falta de recursos del sistema (espacio en disco, memoria)
- Formato de las bases de alias.

Registros relacionados con conectividad.
- Intentos de accesos exitosos y fallidos.
- Problemas de pérdida de comunicación.
- Fallos de los protocolos.
- Expiración de las conexiones.
- Conexiones rechazadas.
- Uso de comandos como VRFY y EXPN.

Registros relacionados con mensajes
- Destinatario.
- Remitente.
- Direcciones de correo malformadas.
- Estadísticas de los mensajes.
- Mensajes congelados.
- Errores en la entrega de mensajes.
- Mensajes rechazados.

Una vez que se ha configurado la información que se quiere registrar, los administradores deben asegurarse que existe suficiente espacio en el disco para almacenar toda la información de los registros ya que éstos aumentan de tamaño de forma acelerada. En ocasiones será necesario comprimir los registros o cambiarlos de lugar.

Conservación y revisión de los registros

Un sistema bien gestionado necesita del registro de sus eventos así como de una completa monitorización de esos registros. La revisión de las actividades del servidor debe hacerse de manera constante (diario). Cuando se descubre alguna anomalía en los registros, entonces se debe hacer una revisión minuciosa para descubrir la causa y corregirla. Otro punto importante es la conservación de los registros ya que

proporcionan un medio para corregir problemas en el servicio y descubrir actividades sospechosas.

Muchas organizaciones respaldan los registros en un servidor central con la finalidad de que si la máquina servidor es comprometida se tengan los elementos para descubrir qué fue lo que paso y entonces tomar las medidas necesarias para que no vuelva a ocurrir.

Herramientas de análisis de registros

La revisión constante de los registros puede ser una tarea muy demandante, más aún cuando el servidor procesa mucha información. Es por ello que existen herramientas que ayudan a los administradores a revisar de forma automática los registros y arrojan un reporte de las anomalías encontradas. Este tipo de herramientas son muy útiles para ahorrar tiempo y tienen buen nivel de configuración por lo que su uso es muy recomendable.

4.7.2 RESPALDO DE INFORMACIÓN

Una de las funciones más importantes de un administrador de correo es mantener la integridad de los datos del sistema. Para lograr la integridad de la información se puede recurrir al respaldo de datos de manera periódica. Es fundamental realizar respaldos de la información de cualquier servidor ya que cualquier eventualidad podría corromper o borrar los datos relacionados con la operación del servidor. Por lo tanto, es necesario tener una política de respaldos en cualquier organización.

La estrategia de respaldos debe contemplar al menos dos métodos: respaldos completos, y respaldos incrementales.

Los respaldos completos implican realizar una copia del sistema de archivos entero, lo que incluye el sistema operativo, aplicaciones y datos localizados en el servidor de correo. La mayor ventaja de los respaldos completos es que resulta fácil restaurar el sistema completo a un estado funcional. La desventaja de los respaldos completos es que consumen gran cantidad de recursos y tiempo, lo que afecta el rendimiento del equipo. Los respaldos incrementales reducen el impacto de tiempo y recursos ya que solo realizan un respaldo base y a partir de él sólo se respaldan los datos que han cambiado. El método elegido dependerá de las necesidades de la organización.

Para tener una política de respaldos adecuada se deben considerar los siguientes puntos:

- Escoger un software adecuado para los respaldos.
- Verificar los datos respaldados.
- Mantener al menos dos copias de los respaldos en diferentes lugares.
- Realizar un calendario acerca de la periodicidad de los respaldos.
- Etiquetar los respaldos, para evitar la duplicidad.
- Tener un plan de recuperación.
- Tener programas que ayuden a automatizar los respaldos y verificarlos.
- Mantener los respaldos en un lugar seguro.
- Realizar los respaldos en distintos medios físicos (cintas, discos duros, CD-ROM, etc.).

4.7.3 RECUPERACIÓN DE UN SISTEMA COMPROMETIDO

En ocasiones es necesario enfrentarse a un ataque exitoso en el servidor de correo. El primer paso para hacer una recuperación del sistema es tener un documento que describa los procedimientos de recuperación antes de que el ataque exista. El documento de procedimientos debe detallar una secuencia de acciones a tomar en caso de un evento de seguridad.

Muchas organizaciones tienen elementos dedicados a la respuesta de incidentes. Estos elementos son los indicados para realizar los procedimientos y es el primer grupo que se debe contactar si existe sospecha o confirmación de alguna anomalía respecto a la seguridad.

Los pasos comunes que se deben realizar después de descubrir que el sistema ha sido comprometido son:

- Reportar el incidente al equipo responsable.

- Tratar de evitar que el daño se propague.

- Realizar alguna prueba en otros equipos de la red para confirmar si han sido vulnerados.

- Análisis de la intrusión:

Capturar el estado actual del servidor (conexiones de red, estado de la memoria, fechas de archivos, fecha y hora del sistema, usuarios conectados, etc.)
Modificaciones realizadas en la configuración de aplicaciones.
Modificaciones realizadas a los datos en general.
Herramientas instaladas.
Registros del sistema, de las aplicaciones y del firewall.

- Restaurar el sistema

– Se tienen dos opciones:
Reinstalar el sistema operativo.
Restaurar a partir de respaldos (esta opción puede ser riesgosa ya que los respaldos tal vez se realizaron después del incidente. Por lo tanto, los respaldos pueden otorgar un acceso al atacante).

– Deshabilitar servicios innecesarios.

– Aplicar todos los parches y actualizaciones.

– Cambiar todas las contraseñas.

– Reconfigurar los elementos de seguridad de la red.

- Realizar pruebas de seguridad y de operación al sistema.

- Reconectar el sistema a la red.

- Monitorear el sistema con la finalidad de ver si el atacante está intentando ingresar nuevamente.

- Documentar la lección aprendida.

Basados en los procedimientos y políticas de la organización, los administradores deben decidir reinstalar el sistema operativo de un equipo comprometido o recuperarlo a partir de los respaldos. La decisión a menudo depende de los siguientes factores:

- Nivel de acceso que el atacante logró (root, user, guest, system).

- Tipo de ataque (interno o externo).

- Propósito del ataque (Enviar Spam, Repositorio ilegal de software, plataforma para otros ataques).
- Método usado para comprometer el sistema.
- Acciones del atacante durante y después del evento.
- Duración del evento de seguridad.
- Propagación del ataque sobre la red (número de equipos comprometidos).
- Opiniones de los expertos en seguridad.

Lo ideal es reinstalar todo el sistema porque nunca se sabe con certeza todas las acciones que realizó el atacante. Aunque, en ocasiones por políticas del funcionamiento del servicio será necesario recuperar a partir de los respaldos con el riesgo que esto conlleva.

4.7.4 PRUEBAS DE SEGURIDAD EN UN SERVIDOR DE CORREO

Pruebas periódicas de seguridad en el servidor de correo son importantes para asegurarse que los mecanismos de protección implementados son los adecuados.

A pesar de que existen diversas técnicas de pruebas de seguridad, dos de las más usadas son: Búsqueda de vulnerabilidades y Pruebas de penetración.

4.7.4.1 BÚSQUEDA DE VULNERABILIDADES

Un escáner de vulnerabilidades es una herramienta automatizada que se usa para identificar vulnerabilidades o configuraciones inseguras en un equipo. Muchas herramientas de búsqueda de vulnerabilidades proporcionan información acerca de cómo corregir las debilidades del sistema.

Las herramientas de búsqueda de vulnerabilidades se basan en bases de datos de para identificar debilidades de seguridad en el sistema operativo y el software instalado. Por ello es necesario actualizar las bases de datos de la herramienta para tener un reporte más confiable.

El escáner de vulnerabilidades es capaz de proporcionar la siguiente información:

- Identificar equipos activos en la red.
- Identificar servicios activos en la red y mencionar los vulnerables.
- Identificar las aplicaciones.
- Identificar el sistema operativo.
- Identificar software sin actualizar.
- Identificar configuraciones que ponen en riesgo la seguridad.
- Identificar los banners inseguros.

Es cierto que las herramientas de búsqueda de vulnerabilidades son muy útiles, pero debe considerarse que también proporcionan falsos positivos, es por ello que un administrador que conozca el sistema debe interpretar todos los resultados arrojados por la herramienta.

El análisis de vulnerabilidades es una labor intensa que requiere una interpretación de resultados por expertos en el tema. Un análisis de vulnerabilidades puede afectar el rendimiento del sistema debido a las tareas que tiene que realizar. A pesar de ello, la búsqueda de vulnerabilidades es importante para mitigar los problemas antes que los atacantes los descubran y los aprovechen.

4.7.4.2 PRUEBAS DE PENETRACIÓN

Son pruebas para intentar ganar acceso a un sistema utilizando técnicas desarrolladas por atacantes y expertos en seguridad. Existen herramientas que ayuda a realizar pruebas de penetración, aunque debe decirse que estas pruebas las deben realizar expertos porque un mal uso de ellas puede dañar el sistema. Las pruebas de penetración tienen impacto en el rendimiento del sistema con lo cual se deben aplicar antes de que el sistema entre en producción o cuando el servicio no tiene tanta demanda.

Las pruebas de penetración ayudan a desarrollar un plan de seguridad para una organización y es recomendable aplicarlas en sistemas complejos y críticos.

Las pruebas de penetración ofrecen los siguientes beneficios:

- Examinan la red usando metodologías y herramientas desarrolladas por atacantes.
- Corroboran la existencia de vulnerabilidades.
- Muestran cómo las vulnerabilidades pueden ser explotadas para ganar acceso.

- Demuestran que las vulnerabilidades no son sólo teóricas.
- Proporcionan ataques reales para implementar medidas serias de seguridad.
- Permiten probar los procedimientos de seguridad de un sistema.

4.7.5 ADMINISTRACIÓN REMOTA DE UN SISTEMA DE CORREO

Se recomienda que la habilitación de la administración remota de un sistema de correo se evalúe realizando un cuidadoso análisis de todos los riesgos. Lo ideal es deshabilitar la administración remota, sin embargo, algunas organizaciones necesitan administración no presencial. Los riesgos de habilitar la administración remota dependen de la localización del servidor de correo en la red. Para servidores que están detrás de un firewall puede ser relativamente seguro acceder a él dentro de la red interna, lo cual no elimina todos los riesgos. La administración remota no debe permitirse desde un equipo localizado fuera de la red local de la organización a menos que se implementen mecanismos de acceso relativamente seguro, como puede ser una red privada virtual.

Si una organización determina que es necesario administrar remotamente un servidor de correo, se deben seguir ciertos pasos para lograr que el servicio esté habilitado de la manera más segura posible.

- Se deben usar mecanismos de autenticación fuertes (Llaves públicas/privadas, criptografía combinada).
- Restringir los equipos desde los cuales se permite la administración remota.
- Usar protocolos seguros que proporcionen cifrado de datos y contraseñas, por ejemplo SSH, HTTPS o IPsec.
- Reforzar el concepto del mínimo privilegio.
- Implementar extensiones de redes locales seguras (VPN).
- Cambiar cualquier cuenta o contraseña que se instalen por default en la herramienta de administración remota.
- No acceder directamente como administrador al sistema.

CAPÍTULO 5

ALTA DISPONIBILIDAD Y SERVICIO DE DIRECTORIOS

5.1 FACTORES QUE DEMANDAN DISPONIBILIDAD

En la actualidad, los profesionales de las tecnologías de la información se ven obligados a trabajar bajo un ambiente complejo donde el éxito de la organización ya no sólo se basa en un solo factor como lo es el mejor equipo o las mejores habilidades técnicas. Ahora, existen diversos factores que influyen en el éxito de una organización de servicios, uno de ellos es la disponibilidad.

Lo que ocurre es que el tiempo tolerable de inactividad se está acortando constantemente, debido al constante aumento de la intensidad de los procesos de negocio, su globalización y necesidad de cooperación en tiempo real entre múltiples unidades internas y externas. También, cada vez mayor número de transacciones se realiza sin soporte en papel, lo cual hace prácticamente imposible su recuperación en caso de que llegue a perderse el soporte informático. Por lo tanto, en los últimos años se ha incrementado la demanda de soluciones de continuidad de servicios, recuperación de desastres y alta disponibilidad de sistemas ya que las organizaciones se han dado cuenta de lo que puede significar que las circunstancias les impidan operar normalmente durante un período de tiempo. Lo que se pretende es asegurar la continuidad de servicios de misión crítica, en otras palabras, aplicaciones que deben estar disponibles de manera permanente y que una interrupción podría acarrear problemas serios para la organización o sus usuarios. Con lo anterior, un sistema de misión crítica debe tener, entre otras, las siguientes características:

- Las aplicaciones que proporcionan el servicio deben poder recuperarse después de un fallo.
- El sistema debe ser configurable sin necesidad de apagarlo.
- El mantenimiento del equipo debe realizarse sin necesidad de interrumpir el servicio.
- Debe existir una tasa baja de fallas en la aplicación.
- El sistema debe ser diseñado para seguir operando en caso de desastres naturales.
- Debe tener una fuente de energía alterna a la habitual.

Para cumplir estas características existen varias soluciones de disponibilidad que persiguen el ideal de proporcionar un servicio permanente, sin embargo, todas las organizaciones saben que no existe

sistema infalible.

5.2 ALTA DISPONIBILIDAD

Disponibilidad es la medida de cuánto tiempo emplea una aplicación o un sistema en hacer un trabajo útil, contra cuánto tiempo gasta fuera de servicio. Disponibilidad no es lo mismo que confiabilidad. Confiabilidad es una medida de la tasa de fallas, disponibilidad se refiere a recuperarse de la falla y continuar. Una aplicación puede estar llena de errores, pero si se recupera de éstos, de forma rápida, todavía puede ser altamente disponible. De otra manera, una aplicación puede estar completamente libre de error, pero todavía no estar 100% disponible. La disponibilidad se refiere a un nivel de servicio proporcionado por aplicaciones o sistemas. Los sistemas de alta disponibilidad tienen un tiempo de inactividad mínimo, ya sea previsto o imprevisto. La disponibilidad se suele expresar como el porcentaje del tiempo que un servicio o un sistema está disponible. Alta disponibilidad **(**High Availability**)** comprende la metodología para proveer un servicio sin interrupciones no planeadas.

5.2.1 INTERRUPCIONES

Un fallo en el sistema puede ser causado tanto por factores internos como externos. Ejemplos de factores internos incluyen, entre otros, los errores de diseño, defectos de manufactura, defectos en los componentes del sistema. Ejemplos de factores externos incluyen la interferencia electromagnética, errores de operación, desastres naturales, etc. Aún cuando un sistema esté muy bien diseñado y los componentes sean confiables, los fallos no pueden ser eliminados completamente. Sin embargo, es posible administrar los fallos para minimizar el impacto negativo en un sistema.

5.2.1.1 INTERRUPCIONES DEL SISTEMA NO PLANEADAS (FALLOS)

Las paradas del servicio no planeadas son el resultado de un fallo inesperado del sistema asociado con un error en los componentes de hardware o de software. Este tipo de paradas son las más costosas ya que producen impactos muy negativos a la organización. En sistemas críticos este tipo de fallas pueden producir resultados catastróficos, por ejemplo, en los aeropuertos pueden causar problemas en el control de tráfico aéreo, en los bancos pueden producir pérdidas millonarias, en los sistemas de electricidad pueden originar “apagones”, etc. En términos generales, las interrupciones del servicio reduce la satisfacción del

cliente.
Algunas causas que producen este tipo de fallas son las siguientes:

- Problemas con el hardware.
- Fallas en la infraestructura de la red.
- Problemas con el sistema operativo.
- Corrupción del Sistema de archivos.
- Dispositivos de almacenamiento a su máxima capacidad.
- Software Malicioso.
- Picos de Energía.
- Ausencia del suministro de Energía.
- Defectos en el firmware.
- Fallas en las aplicaciones que suministran el servicio.
- Desastres naturales y causados por el hombre.
- Errores humanos en la administración del sistema.

5.2.1.2 INTERRUPCIONES PLANEADAS DEL SERVICIO (MANTENIMIENTO)

Las paradas de servicio planeadas deben ser cuidadosamente planificadas para que tengan un impacto mínimo en la disponibilidad del servicio. Las paradas de este tipo son el resultado de eventos de mantenimiento relacionados con la reparación, respaldo u operaciones de actualización del sistema. Las reparaciones intentan remover componentes defectuosos para restaurar el sistema a un estado funcional. Los respaldos tienen como objetivo preservar los datos críticos sobre medios no volátiles y así evitar la pérdida de información cuando un sistema en producción experimenta un fallo en el almacenamiento principal. Las actualizaciones tienen como finalidad el reemplazo del hardware o software con versiones más recientes o versiones mejoradas.

Las interrupciones planeadas generalmente son resultado de las siguientes actividades:

- Cambios en la configuración del sistema.
- Migraciones del sistema.
- Expansión o reparación de Hardware.
- Realización de respaldo.
- Actualización de software.

Las interrupciones planeadas no causan gran impacto si los usuarios son debidamente informados de este tipo de sucesos, de esta manera, los clientes pueden tomar sus precauciones para prescindir del servicio durante el periodo de interrupción. Debe considerarse realizar interrupciones del servicio planeadas durante el periodo en el cual el sistema está más ocioso, como pueden ser los fines de semana, durante la noche, en vacaciones, etc. Sin embargo, existen equipos que dada su naturaleza no pueden estar sin brindar el servicio, para esos equipos existen alternativas para evitar interrupciones planeadas, por ejemplo: respaldos en línea, cambio de hardware en caliente, servidores espejo, componentes redundantes, etc.

5.2.2 DISPONIBILIDAD REAL Y REQUERIDA

Si bien es cierto que un sistema puede sufrir interrupciones en el servicio que provee, también es cierto que los usuarios no siempre se dan cuenta de dichas interrupciones. El tiempo en que un sistema proporciona servicio sin interrupciones no planeadas se le conoce como disponibilidad real del sistema. Por otro lado, el tiempo que los usuarios demandan que el servicio este funcionando se le llama disponibilidad requerida. Generalmente, la disponibilidad se mide desde el punto de vista del usuario ya que son precisamente los usuarios quienes hacen uso del servicio y la mayoría de las veces los afectados directos en caso de alguna interrupción, además si los usuarios no perciben interrupción de un servicio, es como si dicha interrupción no hubiera sucedido (desde su punto de vista) y considerarán al sistema altamente disponible.

En ocasiones, desde el punto de vista del usuario, un servicio se puede considerar no disponible a pesar de que se encuentre en operación.

Un usuario considera un sistema no disponible en los siguientes casos:

- El sistema no se puede acceder. Si los usuarios no pueden acceder al sistema por cualquier razón, el sistema se considera no disponible.

- El sistema es demasiado lento. Un sistema puede estar en operación, sin embargo, si la velocidad en la respuesta es muy pobre, los usuarios se darán por vencidos y considerarán al sistema no disponible.

- El sistema es intermitente. Los usuarios no usaran un sistema si sospechan que el servicio es inestable.

Con lo anterior, la disponibilidad comprende la prevención de interrupciones reales así como interrupciones percibidas por los usuarios. Es por ello, que si un sistema está orientado a proporcionar un servicio, el primer requisito a contemplar es la disponibilidad demandada por los usuarios, y a partir de esa disponibilidad empezar a planear tiempos de mantenimientos del sistema.

5.2.3 CÁLCULO DE LA DISPONIBILIDAD DE UN SISTEMA

El método más conocido para cuantificar la disponibilidad del sistema se basa en el tiempo promedio entre fallos (MTBF, Mean Time Betwen Failures) que experimenta un sistema y el tiempo promedio en que tarda dicho sistema en recuperarse del fallo (MTTR, Mean Time To Recover) y volver a operar normalmente.
La fórmula es la siguiente:

DISPONIBILIDAD= MTBF / (MTBF + MTTR)

Para calcular el tiempo promedio entre fallos se utiliza la siguiente fórmula:

MTBF = TOT / TNOF

MTBF = Tiempo promedio entre fallos (Mean Time Between Failure)
MTTR = Tiempo promedio de recuperación (Mean Time To Repair)
TOT = Tiempo total de Operación (Total Operating Time)
TNOF = Número total de Fallos o Interrupciones (Total No. Of Failures)

Por ejemplo, si se considera una organización que debe proporcionar un

servicio de forma permanente, esto es, las veinticuatro horas, todos los días de la semana y todos los días del año (24x7x365) entonces se tiene:

24 x 365 = 8760 horas

Además, se piensa que tendrá un total de 5 interrupciones al año y de acuerdo a la experiencia se toma alrededor de tres horas restaurar por completo el servicio.
Por lo tanto

TOT=8760
TNOF=5
MTTR=3

MTBF=8760 (horas) / 5 (interrupciones)=1752 horas

DISPONIBILIDAD=1752 / (1752+3)
=0.99829059829059829059829059829
Sacando el porcentaje y redondeando, resulta:

% DISPONIBILIDAD CONTINUA=99.83

El nivel de disponibilidad se acerca al ideal (100%), por lo tanto, muchas veces se toma como parámetro de disponibilidad el número de nueves que una solución de disponibilidad puede ofrecer. Es importante señalar que el paso de un nivel de disponibilidad a su inmediato superior representa una gran mejoría en la operación y como consecuencia el costo también aumenta de forma radical, ya sea en equipo, mantenimiento, personal, etc.

A continuación se presenta una tabla que representa la disponibilidad de ciertos sistemas tomando como parámetro el número de nueves.

Numero de 9s	Disponibilidad	Tiempo Sin Servicio	Tipos de Sistemas
1	**90.0000%**	36 días, 12 horas	
2	**99.0000%**	87 horas, 36 minutos	Impresoras, Fax

3	**99.9000%**	8 horas, 46 minutos	Sistemas no críticos
4	**99.9900%**	52 minutos, 33 segundos	Centros de Datos
5	**99.9990%**	5 minutos, 15segundos	Equipos con arreglos redundantes
6	**99.9999%**	31.5 segundos	Milicia y Aviación

Tabla 5.1 Disponibilidad de acuerdo al número de nueves.

En los sistemas informáticos existen niveles de disponibilidad que (como ya se mencionó) se cuantifican con el número de nueves, pero que están directamente relacionados por el tipo de interrupciones que sufren.

- **Alta Disponibilidad—** Los sistemas o aplicaciones están disponibles durante horas específicas de operación sin fallos planeados.
- **Operación Continua—** Los sistemas o aplicaciones están disponibles 24 horas al día, 7 días a la semana sin paradas no planeadas.
- **Disponibilidad Continua—** Los sistemas o aplicaciones están disponibles 24 horas al día, 7 días a la semana sin paradas planeadas o no planeadas. Este nivel de disponibilidad se demanda en sistemas críticos que proporcionan servicios fundamentales al público en general, por ejemplo, energía eléctrica, sistemas de comunicación, servicios bancarios, sistemas de comercio electrónico, etc. Obviamente, este nivel de disponibilidad es el más difícil y costoso de alcanzar. Los usuarios deben estar conscientes del costo que implica un alto nivel de disponibilidad.

Para cuantificar la disponibilidad ofrecida por cada uno de los conceptos anteriores se procede de la siguiente manera.

A) Se determina el número de horas que el sistema debe estar (idealmente) disponible. Por ejemplo, si se calcula el número de horas durante un mes se tiene:

24 horas al día, 7 días a la semana = 24 horas por día x 7 días = 720 horas por mes.

B) Se determina el tiempo de interrupción sufrida por el sistema durante las horas que debería estar en operación. Si el nivel de disponibilidad que se persigue es “alta disponibilidad”, entonces se consideran únicamente las interrupciones no planeadas. Para una disponibilidad de “operación continua” se consideran aquellas interrupciones planeadas. Finalmente, si la disponibilidad requerida es “disponibilidad continua” se consideran todas las interrupciones.

Por ejemplo, en un mes se tienen dos interrupciones, la primera de ellas duró alrededor de nueve horas debido a un problema en el disco duro (interrupción no planeada). La segunda interrupción la originó el mantenimiento preventivo y duró aproximadamente quince horas (interrupción planeada).

El nivel de disponibilidad alcanzado es:

Disponibilidad = ((A – B)/A) x 100)

A=Horas de Funcionamiento del sistema (idealmente)

B=Interrupciones de acuerdo al nivel de disponibilidad requerido.

Con lo anterior se tiene:

Alta Disponibilidad = ((720 – 15)/720) x 100 **= 97.92%**

Operación Continua = ((720 – 9)/720) x 100 = **98.75%**

Disponibilidad Continua = ((720 – 24)/720) x 100 = **96.67%**

5.3 FACTORES QUE DIFICULTAN LA RECUPERACIÓN DE UN SISTEMA

Cuando se minimiza el tiempo de Recuperación de un sistema después de un fallo, se aumenta de manera considerable la disponibilidad del mismo, sin embargo, no siempre es posible la recuperación inmediata del servicio, ya que existen factores que dificultan esta tarea.

- **Complejidad del sistema—** Entre más complejo es el sistema, más tiempo tomará restablecerlo. Cuando se reinicia un sistema complejo tarda en levantar todos los servicios, así, que cuando ocurre un fallo que obliga a apagar el sistema se prolonga el tiempo de recuperación.
- **Severidad del Problema —** Generalmente, entre más grave es un problema, más tiempo llevará solucionarlo. No es lo mismo reiniciar un servicio que volverlo a configurar.
- **Disponibilidad del Personal—** En ocasiones las interrupciones aparecen después del horario de trabajo del personal. En este caso la interrupción se hace más grande debido al tiempo que le toma

al personal llegar al lugar del problema.

- **Otros Factores—** Aparte de los ya mencionados, existen otros factores que evitan la inmediata solución de una interrupción. Algunas veces un sistema tiene interrupciones largas debido a que no se puede poner fuera de línea porque existe una aplicación ejecutándose. Otros casos involucran la falta de hardware de reemplazo como lo son: fuentes de poder, discos duros, tarjetas de red, etc.

5.4 CONSECUENCIAS DE LA FALTA DE DISPONIBILIDAD

Las organizaciones consideran la alta disponibilidad como una necesidad y no como un lujo ya que aprecian que sus servicios se proporcionen de forma permanente. Una interrupción en la disponibilidad de los servicios puede acarrear problemas. La severidad del problema depende del tipo de servicio ofrecido.

En general las consecuencias de la falta de disponibilidad se resumen a continuación.

- Pérdida de Clientes
- Pérdida de Oportunidades
- Pérdida de Capacidad
- Trabajo perdido o improductivo
- Costos de Restauración
- Penalizaciones
- Mala Publicidad
- Pérdida de Ingresos

En algunas organizaciones el impacto de la falta de disponibilidad puede ser mayor. Por ejemplo, en los hospitales, en los sistemas de navegación o en la milicia.

5.5 PROCEDIMIENTOS PARA MEJORAR LA DISPONIBILIDAD DE UN SISTEMA

Para elaborar un plan efectivo que mejore la disponibilidad de un sistema se debe comprender el funcionamiento del sistema completo y la manera en que cada componente afecta el funcionamiento de todo el

sistema. Se identifican los componentes críticos para así establecer prioridades en su protección. En ocasiones el componente que aparenta ser insignificante causa un gran impacto en la disponibilidad del servicio que proporciona el sistema. Una vez identificado los componentes críticos se buscan los mejores métodos para mejorar la confiabilidad, recuperación, mantenimiento y administración que llevan a una mejor disponibilidad de todo el sistema.

5.5.1 IDENTIFICAR LOS COMPONENTES DEL SISTEMA

Para mejorar la disponibilidad de un sistema, primero se identifica todos los componentes que trabajan juntos para proporcionar cierto servicio. Si un componente importante del sistema tiene altas probabilidades de fallar entonces el servicio que proporciona el sistema tiene altas probabilidades de interrumpir su operación.

La mayoría de los sistemas se dividen en los siguientes elementos:

- **Host o servidor—** Es la parte del sistema donde la mayoría de la información es almacenada y procesada. El servidor es el proveedor del servicio y procesa las peticiones de transacción de datos.
- **Cliente—** Es el componente que hace peticiones al servidor.
- **Red—** Es el componente que permite la comunicación entre el cliente y el servidor. En realidad, engloba todo el equipo asociado a la comunicación.

Para cada uno de los elementos es importante revisar los componentes que los conforman: hardware, software, ambiente en el cual operan, procedimientos y personal.

Hardware es el equipo físico que tiene el sistema. El hardware incluye, entre otros, los siguientes componentes:

- **Unidad Central de Procesamiento—** El dispositivo que controla la operación del sistema informático.
- **Dispositivos de Almacenamiento—** Abarcan medios de almacenamiento volátil o permanente.
- **Dispositivos de Entrada—** Componentes destinados a proporcionar un medio de acceso al sistema, reciben datos o comandos de usuarios, ejemplos de dispositivos de entrada son: ratones, teclados, puertos seriales, etc.
- **Dispositivos de Salida—** Componentes para presentar los datos a los usuarios, pueden ser monitores, bocinas o impresoras.

- **Cables—** Dispositivos que ayuda a conducir la energía hacia los sistemas informáticos.

El Software son los diferentes programas que se ejecutan en el sistema y le permiten realizar sus funciones, el software incluye:

- **Firmware—** Software embebido en los chips del hardware, actúa como interfaz entre los recursos de hardware y el sistema operativos. En las PCs, el firmware se llama BIOS.
- **Sistema Operativo—** Conjunto de programas que se encargan de administrar los recursos de una computadora.
- **Utilidades—** Software que proporciona facilidades de mantenimiento y administración del sistema.
- **Software de Programación—** Software que permite la creación de aplicaciones, incluye lenguajes de programación como C++, Java y PHP, este apartado también engloba a las herramientas de desarrollo como Visual Studio.
- **Aplicaciones—** Programas diseñados para realizar tareas específicas de los usuarios.
- **Middleware—** Programas que ayudan a la comunicación o al intercambio de datos entre varios programas o sistemas informáticos.

“Ambiente” se refiere al equipo externo que necesita el sistema para funcionar de forma adecuada.

- **Energía—** Incluye reguladores automáticos de voltaje, fuentes de energía, UPS, etc.
- **Clima—** Incluye sistemas de enfriamiento para mantener en óptimas condiciones el equipo.

Los procedimientos son las actividades de administración general del sistema, incluye todas las actividades para mantener el sistema en operación.

- **Activación—** Incluye encendido del equipo, habilitación de Servicios y aplicaciones.
- **Operación—** Manejo de recursos, control de entrada/salida manejo de procesos y control de la red.
- **Mantenimiento—** Incluye respaldos y restauración de la información
- **Manejo de Usuarios—** Administración de usuarios y la seguridad.

- **Desactivación—** Parada del sistema, inhabilitación de servicios y aplicaciones.

5.5.2 IDENTIFICAR LOS COMPONENTES CRÍTICOS DEL SISTEMA

Después de identificar los componentes en un sistema, el próximo paso es encontrar los componentes críticos del sistema, aquellos que representan un punto simple de fallo para el sistema. En otras palabras, la tarea es encontrar los componentes que cuando fallan afectan completamente al sistema.

Lo que se pretende es reducir los riegos asociados con los componentes críticos, para ello, se deben considerar los siguientes puntos:

- **Reducir la frecuencia de las interrupciones.**
- **Minimizar la duración de una interrupción.**
- **Minimizar el ámbito de una interrupción—** Reducir las partes del sistema que son afectadas por una interrupción.
- **Prevenir interrupciones futuras.**

5.5.3 ESTABLECER PRIORIDADES

Los sistemas que se deben atender primero para vigilar su disponibilidad deben tener las siguientes características:

- **Alta incidencia de Interrupciones.**.
- **Servicios Críticos.**
- **Gran número de usuarios afectados**
- **Altos Costos de Recuperación**.

5.5.4 ANÁLISIS DE INTERRUPCIONES PASADAS

Cuando se analizan las oportunidades para mejorar la disponibilidad de un sistema, se deben considerar las interrupciones que ha sufrido el sistema a lo largo de su historia. Si se analizan las interrupciones pasadas, se puede aprender de ellas y ganar experiencia. Algunos puntos que se deben analizar son:

- **El origen del Problema—** ¿Qué ocasionó la interrupción? ¿Qué componente del sistema falló o provocó problemas?
- **Recuperación—** Examinar el manejo que se le dio al problema, las circunstancias que envolvieron al problema y los mecanismos que se utilizaron para su solución. El objetivo es educar a todas aquellas personas involucradas en la administración del sistema

para que conozcan las técnicas empleadas para solucionar ciertos fallos.

- **Evitar que fallos similares vuelvan a ocurrir —** Establecer mecanismos para que el mismo problema no se presente dos veces.

5.5.5 IMPLEMENTAR UNA ESTRATEGIA

Para mantener un sistema funcionando de forma continua es necesario llevar a cabo un plan para administrar la disponibilidad del sistema. La estrategia debe incluir los siguientes puntos:

- Estabilizar el sistema actual eliminando todos los problemas existentes. También es importante identificar los problemas frecuentes y usar un análisis de interrupciones para evitarlos en un futuro.
- Implementar procedimientos de administración. Los buenos procedimientos muchas veces determinan en gran medida el éxito en la estabilidad de un sistema. Se debe establecer un plan en la administración de problemas, la administración de cambios y la administración de seguridad.
- Poner especial atención en los sistemas críticos, o en aquellos componentes que causan gran impacto en el sistema si dejan de funcionar.
- Establecer vínculos con otros departamentos especializados dentro de la organización (seguridad, soporte, almacén).
- Utilizar herramientas automáticas de monitoreo para probar la disponibilidad de un sistema.

Finalmente, es importante recordar que la disponibilidad de un sistema empieza con el diseño y la elección de sus componentes.

5.6 TÉCNICAS PARA MEJORAR LA DISPONIBILIDAD DE UN SISTEMA

En la actualidad existen diversas técnicas que permiten mejorar la disponibilidad de un sistema, algunas de ellas son: redundancia, clustering, tolerancia a fallos, aislamiento y automatización de tareas.

5.6.1 REDUNDANCIA

Para mejorar la disponibilidad, hay que implementar mecanismos de tolerancia a errores que enmascaren o minimicen el impacto de las averías de los componentes y las dependencias del servicio. La

tolerancia a errores se logra implementando un sistema de redundancia en componentes de puntos únicos de error.

La redundancia consiste en duplicar un componente o todo el sistema, de manera que se pueda usar ya sea el componente original o el duplicado en cualquier momento. Ya que ambos componentes pueden ser usados, el sistema puede seguir en operación si alguno de ellos falla. De hecho, no existe el impacto en la operación del sistema cuando un fallo ocurre en cualquiera de los componentes replicados. La redundancia mejora los siguientes aspectos:

- **Confiabilidad.** Los fallos son enmascarados, por lo que los usuarios no los perciben, con lo cual el sistema da la apariencia de ser muy confiable.
- **Recuperación.** Las condiciones de error pueden ser corregidas de manera instantánea, ya que el componente duplicado se usa de manera automática en lugar del componente que ha fallado.
- **Mantenimiento.** Cuando ocurre un fallo sobre un componente duplicado del sistema, se puede realizar una reparación del mismo mientras el sistema sigue en funcionamiento, eliminando la necesidad de suspender el servicio.
- **Administración.** A pesar de que al implementar redundancia existen componentes duplicados que administrar, existen menos fallas en el servicio que se necesita administrar. Con lo que el tratamiento de la administración de errores es menor.

5.6.1.1 REDUNDANCIA EN HARDWARE

Fuentes de poder duales pueden operar al mismo tiempo en un servidor de alta disponibilidad. Si alguna fuente falla, la otra continúa en operación y el sistema sigue funcionando con normalidad. Cuando se evalúa la posibilidad de incorporar fuentes de poder duales, es mejor decidirse por aquellas que permiten el intercambio en caliente para que no sea necesario apagar el sistema mientras se intercambia una fuente de poder.

En los servidores de alta disponibilidad se pueden encontrar tarjetas de red duplicadas. Si alguna tarjeta sufre algún desperfecto, se desactiva de forma automática direccionando el tráfico de la red hacia la otra tarjeta. El intercambio sucede casi de forma instantánea con lo que los usuarios no percibirán alguna caída en el servicio.

Arreglos redundantes de disco (Redundant Array of Inexpensive Disks, RAID) combina varios dispositivos de almacenamiento independientes

como si fuera un solo dispositivo, lo anterior se hace con la finalidad de obtener mayor capacidad de almacenamiento, mejor rendimiento o confiabilidad. La forma en que los datos son almacenados en cada uno de los discos lo define el nivel de RAID implementado.

5.6.1.2 REDUNDANCIA DE SOFTWARE

Se pueden duplicar todas las aplicaciones de un sistema para que funcionen sobre un equipo aislado, de esta manera, si el sistema primario deja de operar, la copia estará en condiciones de entrar como sistema sustituto y así evitar la interrupción en el servicio. El intercambio de sistemas debe ser de manera automática y transparente para los usuarios.

Existen dos factores fundamentales en la redundancia como técnica de alta disponibilidad, el primero de ellos consiste en que el componente duplicado debe tomar (de forma instantánea) el trabajo realizado por el componente principal cuando éste deja de funcionar. El segundo factor sugiere que la réplica debe ser exactamente igual que el componente original, con esto se asegura que no habrá problemas de operación si la réplica entra en funciones.

5.6.2 CLUSTERING

La implementación de un clúster permite dividir la carga de trabajo de un sistema ya que el trabajo lo comparten dos o más dispositivos operando al mismo tiempo. Todos los equipos independientes se unen para operar como uno solo, de esta manera, se maximiza la carga de cada equipo tratando de que cada uno tenga una carga de trabajo que depende de cuántos equipos conforman el clúster. Por ejemplo, si el clúster lo forman cinco equipos, al menos, cada componente debe ser capaz de manejar el 20% del trabajo de todo el sistema. Si un componente llega a fallar, los sobrantes deben poder manejar la carga de trabajo extra. Con la implementación de un clúster se mejora el rendimiento de un sistema porque el trabajo se reparte, además, se mejora la disponibilidad ya que sin un nodo falla, los otros responden de manera inmediata para evitar interrupciones del servicio. Realmente existen dos tipos de clúster, el primero de ellos se le llama clúster de alto rendimiento porque su principal objetivo es mejorar el rendimiento del sistema a través de la repartición de cargas de trabajo. El segundo tipo es el clúster de alta disponibilidad que tiene la misión de mantener el servicio operando a través de la replicación de algunos componentes de hardware o software.

Un clúster de alto rendimiento se diseña para que distribuya la carga de manera automática a cada uno de los componentes que lo conforman.

También, debe distribuir la carga si un componente se agrega o se remueve del clúster. Los componentes del clúster se deben colocar en diferentes lugares para minimizar las oportunidades de que un daño físico afecte a dos o más componentes del clúster al mismo tiempo.

5.6.3 TOLERANCIA A FALLOS

La tolerancia a fallos permite que un sistema o componentes de un sistema sigan operando bajo una condición de error o fallo. La mayoría de los sistemas tolerantes a fallos pueden trabajar bajo condiciones de error de manera temporal, es por eso, que se deben establecer acciones correctivas antes que la capacidad de tolerancia a fallos se exceda y como consecuencia falle el sistema en general.

Ejemplos de sistemas tolerantes a fallos son aquellos que se diseñan para operar bajo condiciones climáticas poco comunes, o que pueden operar con niveles de voltajes anormales.

5.6.4 AISLAMIENTO Y PARTICIONAMIENTO

Empleando las técnicas de aislamiento o particionamiento, un sistema se puede dividir en varias unidades, de esta forma, un fallo en una de las unidades no afectará a las otras y se limitarán los errores a una parte del sistema. El principal objetivo de estas técnicas es evitar que el daño se propague a varias partes del sistema.

En general, el aislamiento es físico, se divide las funciones del sistema en dos o más componentes y se les coloca en lugares diferentes. El particionamiento es lógico, se puede seccionar un disco en particiones para que cada una de ellas tenga características diferentes dependiendo de los datos que se pretende almacenar.

Debido a que el aislamiento limita los fallos a ciertas partes del sistema, la recuperación de las interrupciones es más fácil, simple y rápida con lo que se asegura una buena disponibilidad del servicio que proporciona un sistema.

5.6.5 OPERACIONES AUTOMATIZADAS

Las operaciones automatizadas o automatización, es una técnica para reducir o eliminar la intervención humana de ciertas tareas que se deben ejecutar en el sistema. Su objetivo principal es reemplazar los procedimientos manuales con herramientas o programas que pueden automatizar tareas comúnmente realizadas por el administrador para así evitar errores humanos.

La automatización es necesaria en organizaciones donde se procesa mucha información y la cual sería difícil analizar por un humano. Con las operaciones automatizadas se previenen interrupciones del servicio

debido a fallos humanos.

5.7 ALTA DISPONIBILIDAD EN LINUX

Existen varias soluciones para implementar alta disponibilidad en sistemas Linux, por ejemplo, es posible implementar la alta disponibilidad en hardware, replicando todos los componentes del sistema, sin embargo, esta técnica es muy costosa haciéndola inaccesible para organizaciones que no cuentan con un gran presupuesto. Una alternativa a la replicación hardware es implementar una replicación a nivel software sobre equipos distintos, de esta manera, si el equipo primario presenta problemas, el equipo replicado puede entrar en línea evitando que el servicio se vea perjudicado. Esta solución trae un ahorro de costos, ya que el equipo replicado no necesariamente debe ser idéntico en hardware al equipo primario. En Linux las replicaciones a nivel de software se pueden hacer con servidores espejo y monitoreando constantemente el servicio para revisar su disponibilidad.

En Linux existen soluciones de alta disponibilidad a nivel software: LVS (balanceo de cargas y alta disponibilidad), Piranha (Software de alta Disponibilidad ofrecido por Red Hat), UltraMonkey (Balanceo de Cargas y alta Disponibilidad). Las soluciones mencionadas tienen un componte común que le sirve de base para brindar disponibilidad de servicio, este componente se llama Heartbeat.

5.7.1 HEARTBEAT

Heartbeat es el componente principal del proyecto “Linux HA (High Availability)”, proporciona una solución de alta disponibilidad a través de replicación de software revisando la disponibilidad de un sistema y sustituyéndolo en caso de que ocurra un problema. Esta tecnología implementa *heartbeats*, cuya traducción es “*latidos de corazón”*. Funciona enviando periódicamente un paquete (latidos) al servidor primario, en caso de que el paquete no obtenga respuesta se considera que el servidor no está disponible, por lo tanto se sabe que el servidor ha caído y se toman las medidas necesarias. Dichos *latidos* se pueden enviar por un puerto serie, por UDP o por PPP/UDP. De hecho, los desarrolladores de Heartbeat recomiendan el uso de puertos serie porque están aislados de las tarjetas de red.

La implementación de heartbeat permite revisar la disponibilidad de varios recursos, además, facilita la transición automática de equipos en el clúster. En otras palabras, si el servidor primario deja de funcionar se

hace una transición al servidor réplica, el cual toma la IP del servidor primario para continuar proporcionando el servicio. El mecanismo anterior permite que el usuario no se percate de que el servicio sufrió una interrupción.

Heartbeat permite la comunicación entre equipos a través de puertos serie o por conexiones de tarjetas de red, incluso, es posible implementar más de un medio de comunicación a la vez. Los mensajes de Heartbeat se envían por todas las líneas de comunicación al mismo tiempo, así, si una línea de apoyo cae, se avisará de ese problema antes de que la línea principal caiga y no exista una línea secundaria para continuar el servicio.

Heartbeat también se preocupa por la seguridad permitiendo el uso de ciertos algoritmos como CRC, MD5 y SHA1. CRC es el mecanismo que ofrece menos seguridad por lo que sólo es adecuado si la comunicación entre nodos está aislada de la red, SHA1 proporciona una mayor seguridad pero consume más recursos de CPU.

Heartbeat es una solución confiable de alta disponibilidad, al día de hoy se le considera estable, madura y ha sido probada en una gran variedad de servicios entre los que destacan:

- Servidores de Base de Datos
- Aplicaciones ERP
- Servidores Web
- Soluciones de balanceo de cargas
- Servidores de Correo
- Firewalls
- Servidores FTP
- Servidores de Nombres (DNS)
- Servidores DHCP
- Servidores Proxy con caché
- Aplicaciones personalizadas

5.7.1.1 CARACTERÍSTICAS DE HEARTBEAT

- Funciona en todas las distribuciones conocidas de Linux
- Soporta n-nodos como servidores réplica
- Software configurable
- Integración con soluciones completas de alta disponibilidad y alto rendimiento
- Facilidad en la Instalación

- Es OpenSource
- Se puede integrar con interfaces gráficas para facilitar la configuración y la administración
- Permite diversas vías de comunicación entre nodos
- Tiene mecanismos de defensa (STONITH, Shoot The Other Node In The Head) para evitar que dos nodos estén activos con la misma dirección IP
- Permite implementar algoritmos de seguridad en el intercambio de datos
- Software maduro
- Disponible para FreeBSD, OpenBSD y Solaris

5.7.1.2 STONITH (SHOOT THE OTHER NODE IN THE HEAD, ELIMINAR UN NODO)

Uno de los problemas que se pueden presentar en la soluciones de alta disponibilidad es la posibilidad de que un servidor réplica entre en operación mientras el servidor primario todavía está activo, lo que causaría una confusión en la petición de los clientes porque dos equipos tendrían la misma IP. Por ejemplo, si por alguna razón se pierde de manera momentánea la comunicación del nodo primario con el nodo réplica, este último pensará que el servidor principal ha caído y tomará la dirección IP que brinda el servicio, sin embargo, si el servidor primario sigue activo provocará una confusión en las peticiones y en casos más graves una inconsistencia de datos. Para evitar el problema anterior heartbeat implementa lo que se le llama STONITH. Con la técnica STONITH es posible eliminar al servidor primario si se detecta un problema de comunicación. Cuando el servidor réplica envía paquetes y no obtiene respuesta del servidor principal (por un determinado periodo de tiempo) considera que el servidor primario ha muerto y se dispone a tomar su dirección IP, antes de hacerlo, se asegura que el servidor primario no sigue activo enviándole señales para que deje de operar y ceda sus recursos, de esta manera las máquinas clientes no tienen que lidiar con la confusión de hacer peticiones a más de una máquina con la misma dirección IP.

5.8 SERVICIO DE DIRECTORIOS

Las aplicaciones de todo tipo necesitan datos para trabajar. Entre más fácil sea para la aplicación la consulta de datos más eficiente hará su trabajo. Existen varias formas de obtener datos para una aplicación, por ejemplo, se pueden obtener a partir de archivos de texto plano, de bases de datos de tipo relacional o de un “directorio”. Algunos servicios

de red son capaces de manejar información almacenada en diferentes tipos de estructuras. La conveniencia de cada una de ellas depende de la aplicación en particular y de las condiciones del servicio. En algunos casos bastará el uso de texto plano, en muchos otros la solución serán las bases de datos relacionales y en otros casos, será necesario la implantación de servicios de directorios.

Los directorios se diseñaron apara ayudar a las personas a encontrar cierto tipo de información de forma fácil y rápida. Por ejemplo, las personas pueden buscar números telefónicos en el directorio que proporciona la compañía telefónica, se puede ubicar una oficina en cierto edificio a través del directorio dispuesto para tal fin, se conoce la jerarquía de cierto empleado a través del directorio de la organización, etc. Sin los directorios, la búsqueda de información sería muy complicada de encontrar ya que es precisamente la búsqueda uno de los puntos fuertes de cualquier sistema de directorios.

Las ventajas de los directorios son:

- **Ayudan a las personas a organizar la información**. Esto se hace ordenando la información de acuerdo a cierto parámetro. En los directorios telefónicos se hace de acuerdo al apellido paterno, en una organización se hace de acuerdo al puesto jerárquico que se tiene.
- **Los directorios facilitan el manejo de datos.** Un directorio además de ser una forma eficiente de encontrar información, también proporciona mecanismos para su administración.
- **Se pueden centralizar para que sean usados por múltiples entidades.** Al centralizar la información es fácil compartirla por varias entidades, evitando que exista información repetida por cada entidad que desee usarla.

Las características anteriores describen las ventajas de los directorios de una forma muy general. Hablando en el ámbito informático, también existe necesidad de los directorios debido a que existen muchas aplicaciones que necesitan realizar búsquedas en un entorno distribuido. En el contexto de las redes de datos, se denomina *directorio* a una base de datos especializada que almacena información sobre los recursos u "objetos" presentes en la red (tales como usuarios, ordenadores, impresoras, etc.) y que pone dicha información a disposición de los usuarios de la red. Por este motivo, esta base de datos suele estar optimizada para operaciones de búsqueda, filtrado y lectura en lugar de operaciones de inserción o transacciones complejas.

Uno de los protocolos de acceso a directorios mejor conocidos en la actualidad es LDAP (Lightweight Directory Acces Protocol, Protocolo

Ligero de Acceso a Directorio).

5.9 LDAP

LDAP es un protocolo diseñado para acceder a un servicio de directorio, se encuentra en la capa de aplicación de la pila de protocolos TCP/IP.

LDAP permite el acceso a la información del directorio mediante un esquema cliente-servidor, donde uno o varios servidores mantienen la misma información de directorio (actualizada mediante réplicas) y los clientes realizan consultas a cualquiera de ellos. Ante una consulta concreta de un cliente, el servidor contesta con la información solicitada y/o con un "puntero" hacia dicha información o datos adicionales (normalmente, el "puntero" es otro servidor de directorio).

5.9.1 VENTAJAS DE LDAP

- Es muy rápido en la lectura de registros.

- Permite replicar el servidor de forma muy sencilla y económica.

- Muchas aplicaciones de todo tipo tienen interfaces de conexión a LDAP y se pueden integrar fácilmente.

- Dispone de un modelo de nombres globales que asegura que todas las entradas son únicas.

- Usa un sistema jerárquico de almacenamiento de información.

- Permite múltiples directorios independientes.

- Funciona sobre TCP/IP y SSL.

- La mayoría de aplicaciones disponen de soporte para LDAP.

- La mayoría de servidores LDAP son fáciles de instalar, mantener y optimizar.

- Es independiente de la plataforma, funciona en Linux, Solaris, Windows, etc.

- Permite distribuir los datos de manera nativa.

-Es un protocolo estándar, por lo tanto se pueden escribir clientes personalizados a un bajo costo.

5.9.2 ESTRUCTURA DE LDAP

Un directorio se compone de entradas. La entrada es la unidad básica de un directorio. Las entradas contienen información relacionadas con una entidad. Por ejemplo, un directorio podría tener entradas acerca de personas, algunos datos serían: nombre, número telefónico, correo y otra información personal relevante. Entonces, habría una entrada para cada persona y cada entrada consistiría de toda la información conocida por el directorio acerca de la persona. El término “entrada” es sinónimo con el concepto de “registro” o con “objeto de directorio”

Las entradas se componen de un conjunto de atributos

La información asociada a cada entrada se le llama “atributos” o “propiedades” de dicha entrada. Una entrada es básicamente una colección de atributos. El nombre de una persona en una entrada es un atributo, así como también el teléfono de la persona. Dependiendo de la definición del directorio, ciertas entradas pueden tener un conjunto de atributos obligatorios y opcionales.

Un atributo se compone de un tipo de datos y su valor asociado

Cada atributo viene identificado mediante un *nombre* o acrónimo significativo, pertenece a un cierto *tipo* y puede tener uno o varios *valores* asociados. Por ejemplo, cn=Jose Abarca es un atributo, donde cn (common name) es el tipo de atributo, y Jose Abarca es el valor del atributo.

Toda entrada viene identificada unívocamente en la base de datos del directorio mediante un atributo especial denominado *nombre distinguido* o dn (*distinguished name*). El resto de atributos de la entrada depende de qué objeto esté describiendo dicha entrada (`objectclass`).

El atributo `objectclass` define algunas reglas que las entradas deben respetar.

Para todas las entradas existe un atributo especial y obligatorio que se llama `objectclass`. Este atributo determina el contenido que debe tener cada entrada especificando un conjunto de atributos que son obligatorios para cierta entrada y un conjunto de atributos opcionales. El atributo `objectclass` puede contener múltiples valores, en este caso, el conjunto de atributos obligatorios y opcionales será la unión de todos los valores del atributo `objectclass`. Dicho de otra forma, `objectclass` define qué atributos se pueden usar en una entrada.

Schema

El schema define reglas en el directorio.

Así como existen reglas que determinan el contenido que puede tener una entrada (objectclass) también existen reglas que determinan el tipo de entradas que puede tener un directorio. A este último conjunto de reglas se le conoce como “schema”. En otras palabras, schema define los posibles tipos de objetos, así como los atributos (incluyendo su nombre, tipo, valor(es) admitido(s) y restricciones), que pueden ser utilizados por el directorio de un servidor de LDAP. Si un schema no contiene una clase de objeto, no se pueden crear una entrada con esa clase de objeto. Se pueden extender los schema para incluir nuevas clase de objetos o para permitir nuevos atributos opcionales sobre una existente clase de objeto.

5.9.3 MODELOS LDAP

Los modelos LDAP representan los servicios proporcionados por un servidor, tal y como los ve el cliente. Estos modelos describen las diferentes funcionalidades de un directorio LDAP.

5.9.3.1 MODELO DE INFORMACIÓN

El modelo de Información proporciona la estructura y los tipos de datos necesarios para construir un árbol de directorio LDAP.

5.9.3.2 MODELO DE NOMBRES

El modelo de nombres define la forma en que las entradas en el árbol del directorio son referenciadas de forma única. Cada entrada tiene un atributo que es único entre todos los hermanos de un simple padre. Este atributo único se llama nombre distintivo relativo (RDN, relative distinguished name). Se puede identificar unívocamente una entrada dentro de un directorio siguiendo la ruta invertida desde un nodo hasta la raíz del árbol, pasando por todos los RDN’s de las distintas entradas. Este recorrido inverso y que se representa por un conjunto de cadenas es lo que se le conoce como “nombre distintivo” (DN, distinguished name).

Debido a que un directorio almacena información sobre los objetos que existen en una cierta organización, cada directorio posee como raíz (o *base*, en terminología LDAP) la ubicación de dicha organización, de forma que la base se convierte de forma natural en el *sufijo* de los nombres distintivos de todas las entradas que mantiene el directorio.

En la siguiente figura se visualiza un árbol de directorio que ejemplifica una entrada con sus diferentes elementos. Es importante notar que

tanto el nombre del atributo así como su valor se incluyen en el RDN.

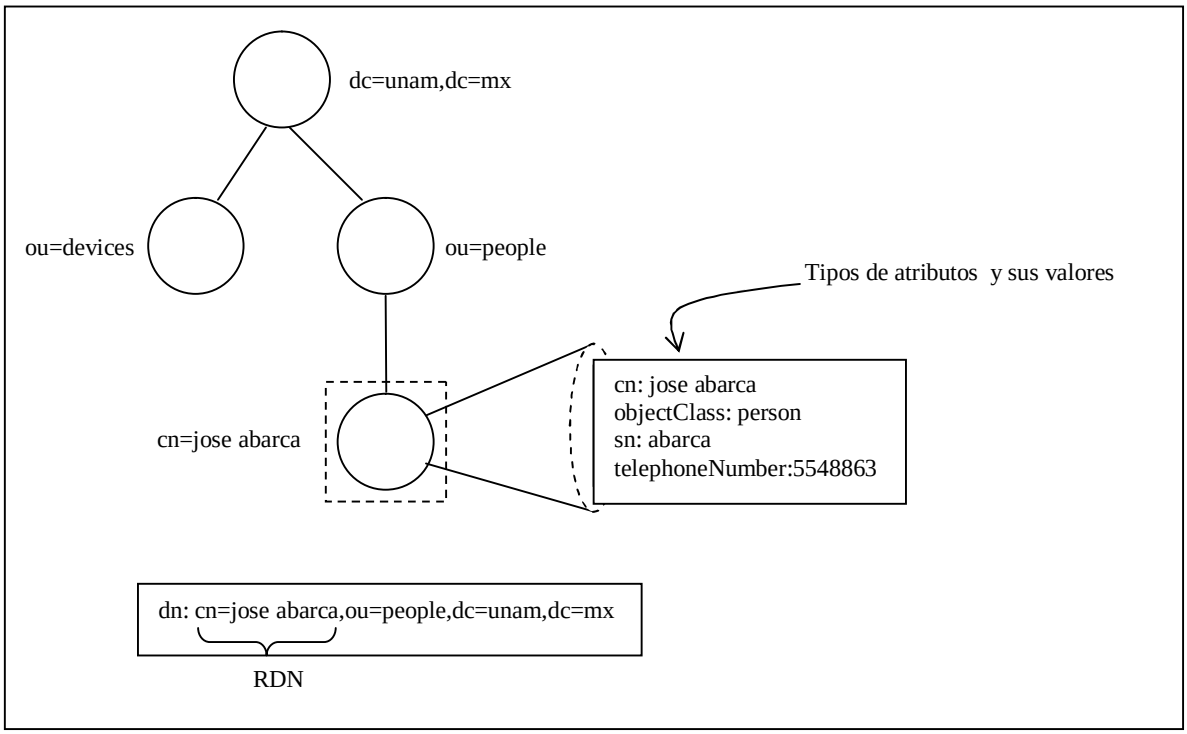


Figura 5.1 Ejemplo de un árbol de directorio LDAP

5.9.3.3 MODELO FUNCIONAL

El modelo funcional es el protocolo LDAP por sí mismo. Este protocolo proporciona los medios para acceder a los datos en el árbol de directorio. El acceso es implementado a través de operaciones de conexión, operaciones de consulta (búsquedas y lecturas) y operaciones de actualización (escritura).

Estas tres categorías de operaciones se muestran en la siguiente tabla.

Operaciones en LDAP	
Categoría	**Tipos de Operación**
Operaciones de conexión	Conexión, desconexión y abandonar
Operaciones de Consulta	Búsqueda y comparación
Operaciones de Modificación	Agregar, modificar, modificar RDN y borrar.
Extensiones	

Tabla 5.1 Operaciones en el protocolo LDAP

Las operaciones de conexión ayudan a controlar la sesión cliente-servidor para todas las subsecuentes peticiones del cliente. La operación de conexión permite al cliente identificarse con el servidor. Esta identificación se usa para que el directorio determine el tipo de autorización y el control de acceso a la información que el cliente tendrá en el directorio. La operación “abandonar” permite que el cliente cancele alguna transacción con el servidor.

Las operaciones de consulta permiten a los clientes explorar el directorio. La operación de búsqueda es la más usada en un directorio y es la operación que tiene más parámetros debido a que es el punto fuerte de un servicio de directorios, los parámetros les permiten a los clientes realizar sofisticadas búsquedas de manera fácil y rápida. La operación de comparación permite a un cliente solicitar la verificación de la información asociada a una entrada. El cliente envía el valor a comparar y el servidor responde de manera exitosa si el valor coincide con una entrada, de otra forma, el servidor reporta que la comparación falló.

Las operaciones de modificación permiten a los clientes cambiar la información contenida en el directorio. Estas operaciones podrían estar restringidas en caso de que el directorio sea de sólo lectura.

También pueden existir operaciones extras que dependen de la implementación del protocolo LDAP, de esta manera, se puede incrementar la funcionalidad de LDAP mediante operaciones conocidas como “extensiones”.

5.9.3.4 MODELO DE SEGURIDAD

El modelo de seguridad proporciona mecanismos que permiten al servidor autenticar a los clientes para que puedan tener acceso a la información almacenada en el directorio. Lo anterior se hace mediante nombres de usuario y contraseñas, además, también es posible determinar el nivel de acceso que cada usuario tiene definiendo controles de acceso, lo anterior se logra normalmente con listas de control de acceso (ACL), pero depende de la implementación de cada fabricante.

Debido a que el directorio opera sobre un ambiente inseguro como lo son las redes, también permite el uso canales encriptados como SSL o TLS, además de otros protocolos de seguridad como SASL.

5.9.4 LDIF

LDIF (LDAP Data Interchange Format) es un formato que permite manipular datos del directorio a través de archivos de texto. Los archivos LDIF se emplean normalmente para realizar operaciones de administración sobre los datos del directorio, por ejemplo, agregar, borrar, modificar, replicar, respaldar datos en el directorio. Al ser archivos de texto es fácil programar scripts para que las manipulaciones de datos se realicen de forma automática aprovechando el formato LDIF. Los datos que se encuentran en el archivo LDIF deben cumplir con las reglas que establece el schema adoptado por el directorio.

Ejemplo de un archivo LDIF.

```
#Archivo LDIF  ***Este es un comentario***
dn: dc=unam,dc=mx
objectClass: domain
dc=unam
```

5.9.5 ¿DÓNDE ALMACENA LA INFORMACIÓN LDAP?

LDAP es un protocolo estándar y no un software que se pueda comprar. Lo que realmente se instala es la implementación del protocolo. El protocolo LDAP no define el lugar donde los datos del directorio deben ser almacenados. Es realmente el fabricante que implementa el protocolo quien decide los diferentes tipos de almacenes que un directorio puede manejar. La mayoría de las implementaciones de LDAP permiten manejar varios almacenes de datos. Es posible usar un manejador de base de datos ligero (LDBM), que también se le llama base de datos embebidas. También se pueden usar archivos de texto plano, e incluso, manejadores de bases de datos relacionales. Ya que LDAP necesita un lugar donde almacenar sus datos se podría confundir con una base de datos relacional. Sin embargo, existen notables diferencias entre un servicio de directorios y una base de datos relacional.

5.10 DIFERENCIAS ENTRE SERVICIOS DE DIRECTORIOS Y BASES DE DATOS RELACIONALES

A continuación se enlistan algunas diferencias entre LDAP y los manejadores de bases de datos.

- Una base de datos proporciona soporte para transacciones, asegurando la integridad de datos. Un servicio de directorios no sabe nada acerca de transacciones por lo tanto no garantiza la integridad en operaciones de escritura ya que no maneja bloqueo de datos.

- El servicio de directorio soporta, en contraste con la base de datos, un complicado mecanismo de replicación, soportando modelos de múltiples maestros o de maestro-esclavo.
- Los directorios están optimizados para operaciones de búsqueda, filtrado y lectura más que para operaciones de inserción o transacciones complejas. Una aplicación que realiza cientos de actualizaciones de datos por minuto no es buena candidata para ser soportada por un directorio. Para este tipo de aplicaciones la solución sería una base de datos relacional.
- Los directorios pueden trabajar en entornos distribuidos. No todas las bases de datos están capacitadas para ello.
- En un directorio es más rápido abrir una conexión y obtener datos que en una base de datos común. Sin embargo una vez abierto el canal de comunicación las bases de datos soportan un conjunto de instrucciones más complejas.

Se puede decir que los directorios son extremadamente útiles si lo que se realiza con frecuencia son las búsquedas y la recuperación de datos. Por lo tanto aplicaciones que manejan datos semi-estáticos son las ideales para ser implementadas junto con un directorio.

Normalmente el tipo de preguntas que se deben responder para saber si LDAP es conveniente para las aplicaciones son:

• ¿Me gustaría que los datos fueran disponibles desde distintos tipos de plataforma?

• ¿Necesito acceso a estos datos desde un número muy elevado de servidores y/o aplicaciones?

• Los datos que se almacenan ¿son actualizados muchas veces?, o por el contrario ¿son sólo actualizados pocas veces?

5.11 USOS DE LDAP

Dadas las características de LDAP sus usos más comunes son:

- **Directorios de información.** Por ejemplo bases de datos de empleados organizados por departamentos (siguiendo la estructura organizativa de la empresa) ó cualquier tipo de páginas amarillas.

- **Sistemas de autenticación / autorización centralizada.** Grandes sistemas donde se guarda gran cantidad de registros y se requiere un uso constante de los mismos.
Por ejemplo:

- Active Directory Server de Microsoft, para gestionar todas las cuentas de acceso a una red corporativa y mantener centralizada la gestión del acceso a los recursos.
- Sistemas de autenticación para páginas Web, algunos de los gestores de contenidos más conocidos disponen de sistemas de autenticación a través de LDAP.
- Sistemas de control de entradas a edificios y oficinas.

- **Sistemas de correo electrónico**. Grandes sistemas formados por varios protocolos que accedan a un repositorio de datos común.

- **Sistemas de alojamiento de páginas web y FTP**, con el repositorio de datos de usuario compartido.

- **Grandes sistemas de autenticación basados en RADIUS**. Control de accesos de los usuarios a una red de conexión o ISP.

- **Servidores de certificados públicos y llaves de seguridad.**

- **Autenticación única para la personalización de aplicaciones.**

- **Perfiles de usuarios centralizados.**

- **Libretas de direcciones compartidas.**

CAPÍTULO 6

REINGENIERÍA DEL SERVICIO DE CORREO ELECTRÓNICO

6.1 REINGENIERÍA DEL SERVICIO DE CORREO DE LA FACULTAD DE INGENIERÍA

La puesta en marcha de un servidor no es tarea sencilla, se debe pasar por varias etapas evaluando en cada una de ellas las mejores opciones de acuerdo a los objetivos que se deseen alcanzar. Por ejemplo, se debe evaluar el sistema operativo a utilizar, la cantidad de servicios que presta el sistema como complemento al servicio principal, el software a utilizar para implantar el servicio, las medidas y herramientas de seguridad para tener una confiabilidad adecuada en cuanto a la información que proporciona el servidor, disponer de los recursos humanos que se encargarán de operar y dar mantenimiento al sistema, establecer procedimientos de respaldo de información así como planes de contingencia y recuperación de desastres, sólo por mencionar algunas. Todo esto parece abrumador, es por ello que se deben establecer pasos ordenados para el establecimiento del servicio.

El presente trabajo sólo se centra en el servicio de correo en sistemas linux, sin embargo la tarea aún es compleja ya que existen muchas distribuciones de este sistema operativo, como consecuencia se deben evaluar aquellas que proporcionen confianza en sistemas críticos.

Cuando se plantee la problemática de rediseñar un sistema de correo en sistemas linux debemos respondernos las siguientes preguntas:

¿Qué distribución se usará?
¿Cuántos usuarios tiene el sistema?
¿Qué servicios adicionales proporciona el servidor?
¿Qué software se utilizará para implementar cada uno de los servicios que tiene el servidor?
¿Las medidas de seguridad que se implementarán?
¿Hardware que se utilizará?
¿Los recursos humanos que operarán y darán mantenimiento al servicio?
Capacitación.

El tema de la distribución a usarse es importante, ya que de ella dependerá en gran medida la vigencia del sistema, uno de los temas importantes de la selección de la distribución es su madurez, soporte y estabilidad. Se deben considerar sólo aquellas orientadas a servidores, que tengan paquetes estables y una comunidad de usuarios amplia para encontrar soporte en caso de necesitarlo. Existen distribuciones comerciales enfocadas al mercado corporativo, dos de ellas que han

logrado posicionarse de manera importante son Red Hat Enterprise Linux (RHEL) y Suse Linux Enterprise Server (SLES). Estas distribuciones ya se consideran una opción en el mundo empresarial gracias a su estabilidad, soporte y una serie de herramientas que proporcionan las soluciones de software que cualquier corporación necesita. Incluso han servido de base para otras distribuciones. Por ejemplo Red Hat es la base del sistema "Unbreakable Linux" impulsada por el gigante de base de datos Oracle. Existen otras que se han ganado su reputación en base al gran número de usuarios y de paquetes, un ejemplo es debian.

La cantidad de usuarios del sistema siempre será un tema fundamental en el diseño de cualquier solución que proporcione un servicio ya que está directamente relacionado a los recursos que se necesitarán en el sistema. Es claro que una gran cantidad de usuarios ocuparán más almacenamiento, lo que afecta la capacidad del disco duro así como el rendimiento del sistema. Así que de inicio la elección de hardware está en función del servicio a proporcionar y el número de usuarios a los que se les proporcionará dicho servicio.

Un sistema puede dar servicios adicionales al principal para complementarlo o dar mayores facilidades en su uso, un ejemplo de ello es un servidor de páginas web que también tiene servicio de transferencia de archivos (ftp) para que sus usuarios intercambien información de forma sencilla. En el caso de correo electrónico muchas veces se requiere de un servidor web como interfaz de correo.

Internet ha traído grandes beneficios al mundo actual pero también ha sido escenario de situaciones indeseables como el robo de información, falsificación de identidades, fraudes, etc. Además se considera un sitio inseguro de intercambio de información, con lo cual, cualquier servidor expuesto a internet debe tener medidas de seguridad. Afortunadamente la seguridad ya es un tema importante en los sistemas informáticos y prácticamente todos los servicios de red cuentan con algún mecanismo de protección (o han sido sustituidos por otros servicios).

En todo sistema informático es vital la intervención de un operador o el administrador, las computadoras no han dejado de ser máquinas que reciben órdenes, por lo tanto el punto medular en el éxito de un sistema es el diseño, la administración y el mantenimiento de dicho sistema, para ello se debe contar con personal calificado para la operación del mismo. El personal debe capacitarse regularmente para perfeccionar el sistema o para tener la información más reciente de los cambios en el

mundo informático y así ser capaces de implantar nuevas funcionalidades en el servidor.

Actualmente, existe un sistema de correo en la facultad de ingeniería que proporciona servicio principalmente a la plantilla de profesores, académicos, tesistas y algunos alumnos de dicha facultad. Lo que se pretende es renovar este sistema haciendo uso de la reingeniería.

JUSTIFICACIÓN DE LA REINGENIERÍA EN EL SISTEMA DE CORREO

Los avances tecnológicos en materia de comunicaciones y específicamente en los sistemas de correo electrónico sugieren un cambio en el servicio de correo que se brinda en la facultad de Ingeniería. En ocasiones no basta con mejorar algunos aspectos del sistema, para tener una mejora importante se debe volver a diseñar el servicio. A continuación se enlistan los motivos que justifican la aplicación de la reingeniería en el sistema de correo de la facultad de ingeniería (UNAM).

- La creciente aparición de software malicioso y propaganda comercial a través del correo lleva a considerar la implantación de ciertos mecanismos para disminuir este tipo de elementos.
- El espionaje en la red obliga a utilizar protocolos que dificulten la traducción de la información.
- La demanda de la continuidad de servicio muestra la necesidad de realizar un diseño de disponibilidad en el servicio.
- La tendencia en el envío de contenido diverso origina la necesidad de aumentar el espacio en los buzones de correo.
- La cantidad de información que arroja un servidor es considerable, por ello se necesitan implantar herramientas que faciliten el análisis de dicha información.
- La cantidad de usuarios que maneja el servidor hace complicado la administración del sistema.

Las necesidades citadas obligan a una renovación del servicio, para ello se utilizará la reingeniería.

6.2 METODOLOGÍA

Ahora que se ha tomado la decisión de aplicar reingeniería al sistema de correo electrónico de la facultad de ingeniería surge la pregunta ¿Qué metodología usar?, varios autores opinan que no existe una metodología clara para llevar a cabo una reingeniería, sino que la metodología depende de las circunstancias del equipo. También se necesita decidir si se va a implantar el servicio nuevo en paralelo al antiguo o sustituirlo completamente.

La mala fama de la metodología

Mucha gente gusta de trabajar de forma libre, por lo que es común notar cierta resistencia al uso de métodos de trabajo, que impliquen un orden específico ya que con ellos se tienen restricciones indeseables sobre todo cuando existe una serie de reglas que dictan lo que hay que hacer, cómo se tiene que hacer y cuándo. Además, algunas personas consideran a las metodologías faltas de imaginación porque imponen un proceso paso y se cree que no ofrecen la oportunidad de pensar libremente.

6.2.1 SELECCIÓN DE LA METODOLOGÍA

A pesar de la resistencia de muchas personas a seguir una metodología, éstas ofrecen un medio ordenado, fomentan la disciplina e imponen puntos de revisión para el seguimiento de determinado proyecto. Esto no quiere decir que las metodologías sean milagrosas y capaces de realizar cambios por sí mismas.

En la reingeniería existe una metodología que no es estricta, fomenta el pensamiento en lugar de suprimirlo y busca el cambio radical. A esta metodología se le conoce como “Rápida Re”, en dicha metodología el equipo de reingeniería se ve precisado en entender, pensar y cuestionar los asuntos tales como:

- Estrategias del servicio.
- Expectativas y percepciones de los clientes.
- Aspectos de valor agregado de los servicios.
- Potencial de cambio radical.
- Deficiencia de los servicios actuales.
- Visión de lo que puede ser si se satisfacen las expectativas del cliente y se eliminan las deficiencias.
- Oportunidades de combinación e integración de los servicios.

- Capacitación de personal.
- Alternativas de implementación.

Debido a las razones anteriores la metodología seleccionado es la "Rápida Re".Esta metodología se usa en la reingeniería de procesos por lo cual se tratará de adaptar en la reingeniería del servicio de correo electrónico.

La metodología "Re" cuenta con cinco etapas:

1**. Preparación.** Se realiza la búsqueda de metas, se forma al equipo y se definen los puntos que justifican la reingeniería.

2. **Identificación**. Desarrollo de un modelo orientado al cliente.

3. **Visión**. En esta etapa se desarrolla una visión del proceso capaz de producir un avance decisivo en rendimiento.

4. **Solución**. Se realiza un diseño técnico basándose en la visión (etapa 3).

5. **Transformación**. Implementa las visiones, aplicando el diseño de la etapa de solución.

6.2.2 PREPARACIÓN

El correo electrónico en la facultad es uno de los servicios de red más utilizados por lo que debe mantenerse funcionando la mayor parte del tiempo, su diseño debe ser moderno y debe tener la capacidad de escalar el número de buzones sin tener un impacto importante en el rendimiento, además debe contar con software que facilite su administración debido al gran número de usuarios que maneja y a la cantidad de correos que procesa diariamente.
Las metas principales es que el correo se mantenga funcionando la mayor parte del tiempo posible, teniendo sólo interrupciones programadas a causa de mantenimiento, respaldo, reinstalación, etc.
Algunos objetivos que se persiguen con la reingeniería del sistema se muestran a continuación:

- Tener un servicio rápido y eficiente.
- Contar con medidas de seguridad modernas.

- Facilitar la administración del sistema y tener procedimientos adecuados para su operación.
- Implementar mecanismos de alta disponibilidad.

El equipo de reingeniería son los administradores del correo de la unidad de servicios de cómputo académico (UNICA) teniendo como líder al jefe del departamento de redes de la misma unidad.
Una de las razones para rediseñar el sistema de correo, es el cambio tecnológico frecuente que obliga a un rediseño de los sistemas para estar a la vanguardia con los mejores sistemas de correo.

6.2.3 IDENTIFICACIÓN

Este apartado se relaciona con un modelo orientado al cliente. Aquí es importante resaltar que el cliente no sólo son los usuarios a los que se les presta servicio de correo, sino también los operadores del servidor, ya que la intención es mejorar el sistema de correo así como facilitar su operación y mantenimiento porque de ello depende una respuesta rápida en caso de que suceda algún incidente.
Obviamente la tarea principal del sistema es la entrega y el envío de correo de forma confiable, pero otros puntos importantes son la privacidad, la rapidez, el espacio de los buzones de correo y la facilidad de uso. Todos estos puntos serán contemplados en el diseño del servicio como beneficio para los usuarios del sistema. Por otra parte los administradores demandan mayor facilidad en el manejo del sistema, la instalación de herramientas que automaticen tareas tediosas como lo son la revisión de bitácoras, la instalación de los paquetes, revisión del rendimiento del sistema, etc.
Las principales demandas de los usuarios se refieren al espacio en el sistema y la privacidad en el correo electrónico

6.2.4 VISIÓN

Para que se logre el objetivo de lograr una mejora importante en el sistema de correo se deben analizar los elementos principales que constituyen el sistema y evaluar cada uno de los paquetes diseñados para cumplir esta función a fin de elegir sólo aquellos que nos permiten ofrecer el servicio de acuerdo a las necesidades de los usuarios y administradores y que se puedan instalar fácilmente en la infraestructura que ya se tiene. En otras palabras lo que se pretende es hacer un “traje a la medida”, maximizar las prestaciones con los recursos que se cuenta.

6.2.5 SOLUCIÓN

Antes de rediseñar un sistema de correo se debe tener registro del número de usuarios con los que cuenta el sistema, la cantidad de correos que se procesan al día, servicios adicionales, y el posible crecimiento del sistema para que no quede desfasado en poco tiempo
En el caso particular del sistema de la facultad de ingeniería se registró durante un semestre la cantidad de correo que procesa el sistema diariamente, sin contar fines de semana ni vacaciones (periodo en el que hay un decremento significativo de la actividad del sistema). El promedio de los correos procesados se eleva a poco más de mil, con lo que el nuevo sistema debe ser capaz de manejar una cantidad de mensajes mucho mayor.

En este punto iniciaremos con la revisión de las principales características del sistema actual para empezar a rediseñar el sistema de acuerdo a la información obtenida.
Debido a las necesidades del correo en la facultad se tienen tres protocolos relacionados directamente con el envío y recuperación de los mensajes, además se cuenta con scanner de virus y spam, una interfaz de correo web y servicio de shell remota.

Sistema de correo actual

Cantidad de usuarios	7500
Cantidad de mensajes mensuales	45000 (aprox.)
Servicios que presta	SMTP, PO3, IMAP, SSH, Webmail
Almacenamiento por usuario (default)	10MB
Formato de los buzones	Mailbox
Características de seguridad	Antivirus, Antispam,
Herramientas de monitoreo	Bitácoras y comandos

Un shell remoto no es necesario en el funcionamiento del sistema de correo, además adiciona carga extra al servidor al ocupar procesos del mismo y aumenta las probabilidades de incidentes de seguridad al permitir que los usuarios interactúen directamente con los comandos del servidor. Todo esto sin contar que representa un servicio más que administrar para los operadores del sistema con todo lo que ello implica, como es revisión de bitácoras supresión de comandos, cambio de permisos, etc.
Estas razones llevan a considerar la separación del servicio de shell remota del servicio de correo, esto no quiere decir que los usuarios ya no cuenten con este servicio, simplemente se trata de buscar un

mecanismo para evitar cargas en el sistema sin afectar las actividades de los usuarios en el sistema. Lo más adecuado es implementar los servicios de correo y el servicio de shell en máquinas diferentes. Este último caso provoca que los usuarios ya no interactúen de forma directa con el servidor lo que sugiere utilizar usuarios virtuales en lugar de usuarios del sistema. De esta manera los usuarios sólo tendrán acceso a las funciones exclusivas del correo, con lo que se reducen las tareas de administración y aumenta la seguridad del sistema.

Para implementar usuarios virtuales se requiere un sistema de autenticación ajeno al sistema operativo, existen dos métodos comúnmente utilizados para este fin, uno de ellos es la autenticación mediante alguna base de datos relacional y el otro es la utilización de un protocolo de directorios como es LDAP (en ocasiones es la combinación de ambos). Como se pretende ahorrar recursos en el servidor es más adecuado implementar la autenticación mediante LDAP porque es un método más ligero y optimizado para búsquedas, tanto así que se emplea no solamente como método de autenticación sino también como método centralizado de información en redes muy grandes.

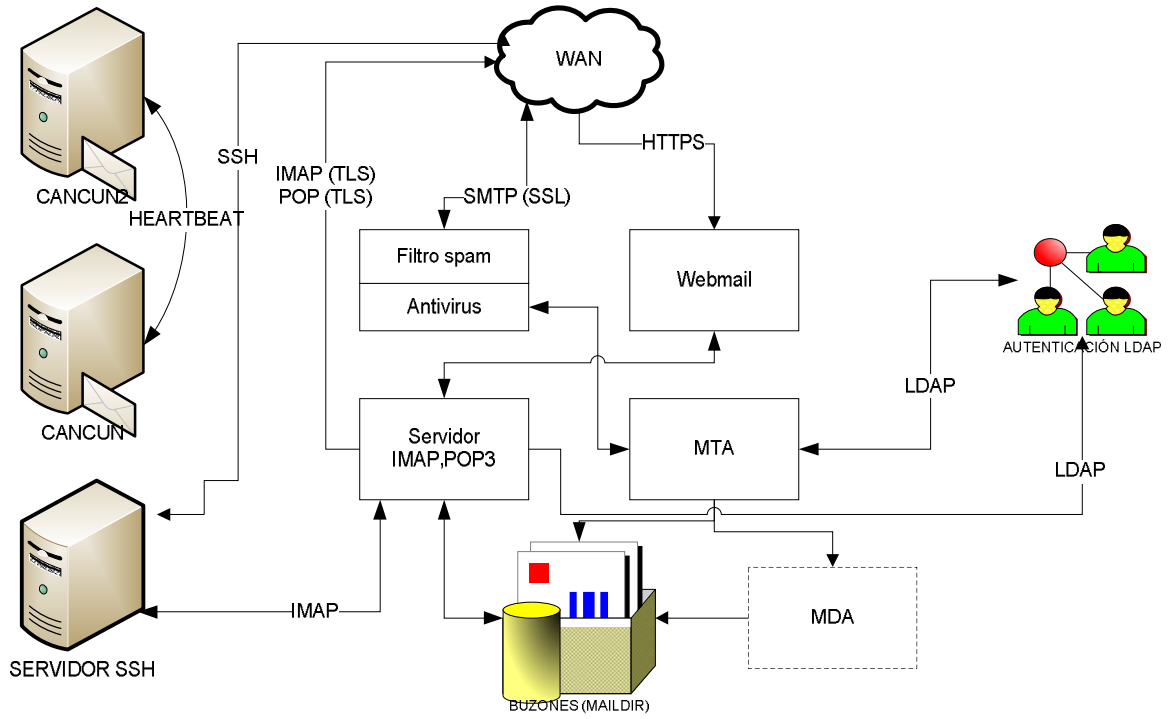


Figura 6.1 Diseño general del sistema de correo

La figura 6.1 muestra el diseño del sistema de correo de una manera general. Los correos entrantes al servidor revisan el destinatario y en caso de que se encuentre un usuario válido del sistema (a través de LDAP) se desvían hacia un scanner que contiene filtros de spam y antivirus, la información del destino de los buzones se encuentra en el servicio de directorio (LDAP) y, cuando sea necesario, serán filtrados por un agente de entrega de correo (MDA). Las conexiones a través del webmail recuperan correos por medio del protocolo IMAP y envían correo inyectando mensajes al agente de transferencia (MTA), después, el agente direcciona los mensajes hacia el scanner para después enviarlos a su destino. Todas las conexiones mediante web son encriptadas para mayor seguridad.

Las conexiones a través de POP3, IMAP se pueden realizar de forma encriptada por medio de SSL (TLS). El servidor de acceso remoto ya no estará en la misma máquina que ofrece el servicio de correo, a pesar de esto, los usuarios podrán revisar su correo como lo vienen haciendo hasta ahora, a través del protocolo IMAP y utilizando la misma aplicación (PINE). También, se pretende instalar un servidor “espejo” para proporcionar un servicio de alta disponibilidad.

6.2.5.1 SELECCIÓN DEL SOFTWARE

En un inicio se debe decidir la distribución de linux en la cual se alojará el servicio, aquí podría surgir un debate muy grande entre seguidores de cada una de las distribuciones, sin embargo, los elementos que operan el sistema se capacitan para operar distribuciones derivadas de Red-Hat (distribución enfocada al mercado comercial, se pagan licencias de uso) con lo que es conveniente buscar una distribución compatible con la mencionada.
Existen dos distribuciones populares y con gran soporte que derivan de Red-Hat: Fedora y CentOS (**C**ommunity **ENT**erprise **O**perating **S**ystem). Éstas distribuciones son prácticamente iguales en su operación, lo que cambia son los paquetes incluidos y las versiones de los mismos. De hecho, las distribuciones de Linux tienen tres grandes diferencias: la instalación, el sistema de paquetes (distribución y manejo de software) y la administración del sistema (herramientas, archivos, inicio del sistema, etc.).
Fedora es una distribución que se le ha denominado "experimental" debido a que incluye las últimas versiones de los paquetes, cambia frecuentemente sus paquetes y sirve como prueba para los paquetes que finalmente constituirán a Red-Hat.
CentOS es una distribución orientada a los servidores ya que contiene compatibilidad binaria con Red-Hat pero sin tener que pagar licencias, por lo que su uso no representa costo alguno, contiene estabilidad en paquetes y una compatibilidad con casi todas las arquitecturas que soporta Red-Hat.
La distribución que se utilizará en el sistema de correo es CentOS 5.2.

6.2.5.2 AGENTE DE TRANSFERENCIA DE CORREO

El protagonista de un sistema de correo es el agente de transferencia de correo y la elección de este elemento es clave en el correcto funcionamiento del servicio. En capítulos anteriores se repasaron las características de cuatro sistemas de correo (qmail, sendmail, postfix y exim) que han sido implementados con éxito como agentes de correo en muchos sistemas de producción, los cuatro tienen una gran comunidad de usuarios, por lo que la decisión se basará en el rendimiento, la facilidad de uso y la seguridad.

Se realizaron pruebas de rendimiento de cada uno de los agentes de correo, estas pruebas son "out-box", es decir, se instalo el software y se le hicieron las configuraciones mínimas para su funcionamiento (ninguna

optimización), las evaluaciones hechas se realizaron en las mismas condiciones y en la misma máquina.

Tomando en cuenta que se procesan más de mil correos al día (en promedio), la prueba consistió en inyectar mil correos y registrar qué sistema mostraba un mejor comportamiento. Se utilizó la herramienta "smtp-source" que permite inyectar correos y añadirle parámetros importantes, también se ocupó una utilidad del sistema para medir el tiempo que lleva a cabo cierto proceso, esta herramienta es "time".

Algunas opciones de smtp-source son:
-c muestra la cuenta de los mensajes enviados.
-l "número" Permite especificar el tamaño del mensaje
-m "número" Cantidad de mensajes a enviar
-f remitente
-t destinatario

La prueba consistió en inyectar mensajes con smtp-source y medir el tiempo que tomó procesarlos con "time", para esta prueba se inyectaron 200 correos tomándose medidas de tres eventos para cada agente, después se hizo la misma prueba con 1000 correos.

Prueba con 200 correos

```
# time smtp-source –c –l 1000 –m 200 –f alfredo@localhost –t instalacion@localhost localhost
```

	Sendmail	Postfix	Exim	Qmail
1	23.639 seg	10.617seg	30.802 seg	15.16 seg
2	23.672 seg	9.953 seg	30.235 seg	14.983 seg
3	22.93 seg	10.38 seg	30.503 seg	15.3 seg

Prueba con 1000 correos

```
# time smtp-source –c –l 1000 –m 1000 –f alfredo@localhost –t instalacion@localhost localhost
```

	Sendmail	Postfix	Exim	qmail
1	1m 59.979 seg	49.974 seg	2m 34.334 seg	1m 15.345 seg
2	2m 3.235 seg	52.137 seg	2m 32.668 seg	1m 17.285 seg
3	2m 2.45 seg	51.543 seg	2m 32.803 seg	1m 17.3 seg

Los resultados muestran que postfix tiene un mejor comportamiento en el procesamiento de correo, además de ser un software muy sencillo de configurar y tener la seguridad como uno de sus puntos fuertes.
Por lo anterior y por las características repasadas en el capítulo dedicado a revisar las características de los agentes de transferencia de correo, se decidió implementar “postfix”.

6.2.5.3 SERVIDOR POP3 E IMAP

Existen dos aplicaciones a considerar en este apartado: Cyrus-IMAP y dovecot. Ambos cuentan con características deseables en un servidor moderno: seguridad, velocidad, varios formatos en los buzones, etc. Sin embargo, la administración en Cyrus es más complicada debido a que se necesita aprender nuevos comandos para su operación. Para sistemas muy grandes (más de 30 000 cuentas) Cyrus sería la mejor opción ya que permite balanceo de cargas. En este sistema se implementará dovecot ya que es un software seguro, ligero y muy rápido, además es fácil configurarlo y administrarlo.

La información de los usuarios (autenticación, ubicación de los buzones, quota, etc.) estará almacenada en un servicio de directorio, en este caso se utilizará el protocolo LDAP. El software que proporcionará este servicio es OpenLDAP ya que es un sistema maduro, soporta replicación, tiene una gran comunidad de usuarios y soporta encriptación. Existe un software con más prestaciones para el servicio de LDAP: Fedora Directory Server, ésta aplicación contiene una consola de administración avanzada, sin embargo, es un software que demanda más recursos lo cual no es ideal en el servidor de la facultad.

6.2.5.4 ADMINISTRACIÓN DE LDAP PARA EL CORREO ELECTRÓNICO

Debido a que LDAP es un protocolo que no se había manejado anteriormente en el sistema de correo, se pretende implementar un sistema que facilite el manejo de las cuentas de correo a través de un ambiente gráfico, para ello se va instalar una consola de administración de correo para LDAP llamado Phamm. Este sistema es muy ligero y automatiza muchas tareas del manejo de LDAP en el correo electrónico (también maneja varios módulos para implementar LDAP en otros servicios).

6.2.6 Transformación

En esta etapa se hará la implementación de la etapa de solución en un servidor de pruebas, la figura 6.2 muestra el diagrama de bloques de la implementación.

El sistema operativo instalado es CentOS 5.2, antes de proceder a la instalación de los paquetes que proporcionarán el servicio de correo, se procede a realizar un "hardening" (configurarlo para reforzar la seguridad) del sistema, a continuación se enlistan los puntos importantes:

- La instalación se realizó de manera personalizada, es decir se instalaron sólo aquellos paquetes necesarios para el funcionamiento del sistema (no se instaló el ambiente gráfico).
- Se actualizaron los paquetes.
- Se desactivaron servicios innecesarios.
- Se deshabilitaron las cuentas del sistema.
- Se configuró un firewall de host.
- Se instaló portsentry una herramienta para evitar el barrido de puertos.
- Se instaló una herramienta de detección de intrusos (AIDE).

Un paso previo a la instalación del sistema de correo, es la creación de la cuenta de los usuarios virtuales.

```
#groupadd vmail
#useradd -g vmail -d /home/vmail vmail
```

Crear directorio donde estarán almacenados los correos de los dominios:

```
#mkdir ~vmail/domains
#chown vmail.vmail ~vmail/domains
```

Crear directorio para el dominio principal:

```
#su - vmail
$ mkdir -p /home/vmail/domains/cancun2.fi-a.unam.mx
```

La siguiente figura muestra el orden de la instalación de los paquetes.

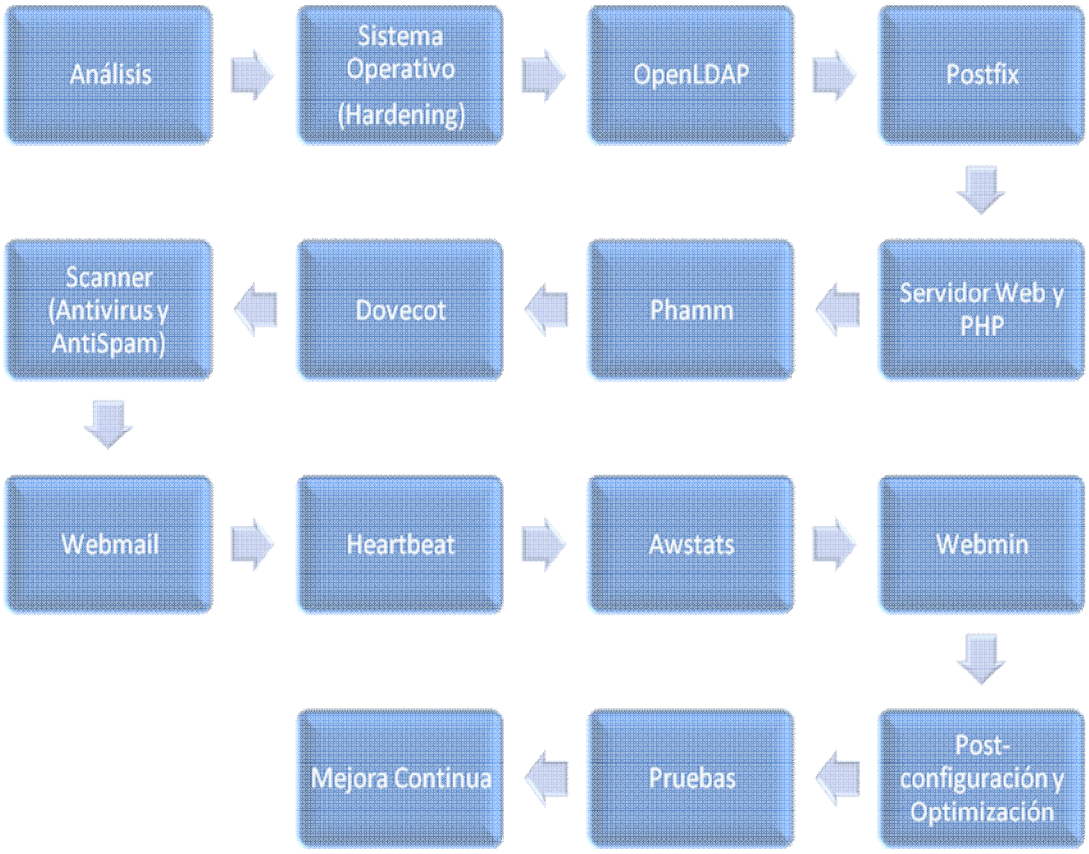


Figura 6.2 Instalación de Software

6.2.6.1 OpenLDAP

El almacenamiento de la información de LDAP se guardará en db-berkeley, por lo tanto, primero instalaremos este paquete.

```
#tar -zxvf db-4.5.20.tar.gz
#cd db-4.5.20
#../dist/configure
#make
#make install
```

Para instalar OpenLDAP

```
#tar -zxvf openldap-stable.tgz
#cd openldap-version/
#env LDFLAGS="-L/usr/local/lib -L/usr/local/BerkeleyDB.4.5/lib -L/usr/lib
-L/usr/local/lib -L/usr/lib/sasl2" \
CPPFLAGS="I/usr/local/BerkeleyDB.4.5/include \
-I/usr/local/include -I/usr/include/sasl" \
 ./configure --prefix=/usr/local/ldap --enable-crypt
#make depend
#make
#make test
```

```
#make install
```

Las opciones en el comando “configure” de LDAP son para indicarle la ubicación de las librerías así como de los archivos de cabecera de db-berkeley.

Para la configuración de OpenLDAP con el correo se necesita el paquete phamm.

```
#wget http://open.rhx.it/phamm/phamm-0.5.10.tar.gz
#wget http://open.rhx.it/phamm/phamm-0.5.10.tar.gz.sig
```

Se revisa el archivo md5 para evitar paquetes corruptos

```
#md5sum -c phamm-0.5.10.tar.gz.sig
#tar -zxvf phamm-0.5.10.tar.gz
#cd phamm-0.5.10
# cd schema/
# cp amavis.schema,ISPEnv2.schema,phamm.schema \
/usr/local/ldap/etc/openldap/schema/
```

Agregar los schemas al archivo de configuración de openldap

```
#vi usr/local/ldap/etc/openldap/slapd.conf

########################################################################
# Schemas
########################################################################
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/nis.schema
include         /etc/openldap/schema/phamm.schema
include         /etc/openldap/schema/ISPEnv2.schema
include         /etc/openldap/schema/amavis.schema
```

Agregar al mismo archivo las listas de control de acceso, se encuentran en phamm;

```
include         /usr/local/ldap/etc/openldap/phamm.acl
loglevel        256  #información de logs de ldap
allow bind_v2


 #cp /usr/local/src/phamm-0.5.10/examples/phamm.acl\
/usr/local/ldap/etc/openldap/
```

El archivo anterior es de ejemplo y es necesario cambiar **dc=example,dc=tld** al nombre de nuestro servidor.

El siguiente paso es agregar la definición de la base de datos y especificar al usuario Manager de openldap, para hacerlo se edita el archivo de configuración “slapd.conf”.
El password encriptado se crea utilizando el siguiente comando:

```
#/usr/local/ldap/sbin/slappasswd
New password: PASSWORD
Re-enter new password: PASSWORD
```

El password creado se copia y se pega en el archivo slapd.conf, en el apartado rootpw

```
#######################################################################
# BDB database definitions
#######################################################################

database    bdb
suffix            "dc=cancun2,dc=fi-a,dc=unam,dc=mx"
rootdn            "cn=Manager,dc=cancun2,dc=fi-a,dc=unam,dc=mx"
# Cleartext passwords, especially for the rootdn, should
# be avoid.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw            {SSHA}i1Uk9QQqRzYav0RT/tNvjhZUgDK0HAVx
```

En este mismo archivo se pueden definir índices para agilizar las búsquedas, los recomendables para el sistema de correo son los siguientes:

```
# Indices to maintain
#index      objectClass                 eq
index   objectClass                   eq
index   cn,mail                       eq,subinitial
index   vd,delete                     eq,pres
index   accountActive,forwardActive   eq,pres
index   smtpAuth                      eq,pres
index   associatedDomain              pres,eq,sub
index   aRecord                       pres,eq
index   sn,uid,displayName            pres,eq,sub
index   uidNumber,gidNumber           eq
index   default                       sub
```

Ahora es momento de crear los nodos superiores de ldap, una plantilla útil se puede encontrar en phamm.

```
#cp /usr/local/src/phamm-0.5.10/examples/sample-main.ldif\
/usr/local/ldap/etc/openldap/nodosraiz.ldif
```

Editar el archivo para que contenga información del servidor
#vi `/usr/local/ldap/etc/openldap/nodosraiz.ldif`

```
dn:dc=cancun2,dc=fi-a,dc=unam,dc=mx
objectClass: top
objectClass: domain
dc: cancun2

dn:cn=Manager,dc=cancun2,dc=fi-a,dc=unam,dc=mx
objectClass: top
objectClass: organizationalRole
cn: Manager
```

Para agregar la información a ldap, es necesario que el servidor se ejecute

```
#/usr/local/ldap/libexec/slapd -h ldap://127.0.0.1
```

Dar de alta la información:

```
#ldapadd -v -x -D "cn=Manager,dc=cancun2,dc=fi-a,dc=unam,dc=mx" -W –h\
localhost -f /usr/local/ldap/etc/openldap/nodosraiz.ldif
ldap_initialize( ldap://localhost )
Enter LDAP Password: PASSWORD
```

Ya que se tienen los nodos superiores, se puede insertar información referente a los usuarios del correo, la plantilla general de phamm puede ser de mucha utilidad

```
#cp /usr/local/src/phamm-0.5.10/examples/sample-mail.ldif\
/usr/local/ldap/etc/openldap/usuario.ldif
```

Este archivo es sólo una plantilla (se tienen que cambiar los datos).

```
dn:cn=phamm,o=hosting,dc=example,dc=tld
objectClass: top
objectClass: organizationalRole
objectClass: simpleSecurityObject
cn: phamm
userPassword: {SSHA}zCfpI3LEnzO0ZWJo8VeHes6TicnPJuME

dn:vd=example.tld,o=hosting,dc=example,dc=tld
objectClass: top
objectClass: VirtualDomain
postfixTransport: virtual:
lastChange: 1066742030
vd: example.tld
delete: FALSE
accountActive: TRUE
adminID: cn=matrix,ou=admin,dc=example,dc=tld
editAV: FALSE
maxAlias: 20
maxMail: 10
maxQuota: 250
```

```
dn:cn=postmaster,vd=example.tld,o=hosting,dc=example,dc=tld
objectClass: top
objectClass: VirtualMailAlias
mail: postmaster@example.tld
lastChange: 1066742031
maildrop: postmaster
accountActive: TRUE
cn: postmaster
sn: postmaster
userPassword: {SSHA}G7vRIKNRvDQg0T0qarcMgbYaQ+iOLEgq
editAccounts: FALSE

dn:mail=abuse@example.tld,vd=example.tld,o=hosting,dc=example,dc=tld
objectClass: top
objectClass: VirtualMailAlias
mail: abuse@example.tld
lastChange: 1066742031
maildrop: postmaster
accountActive: TRUE
cn: abuse
sn: abuse

dn:mail=john.doe@example.tld,vd=example.tld,o=hosting,dc=example,dc=tld
objectClass: top
objectClass: VirtualMailAccount
objectClass: Vacation
objectClass: amavisAccount
objectClass: VirtualForward
mail: john.doe@example.tld
vdHome: /home/vmail/domains
mailbox: example.tld/john.doe/
delete: FALSE
accountActive: TRUE
sn: Doe
userPassword: {SSHA}G7vRIKNRvDQg0T0qarcMgbYaQ+iOLEgq
description: Vacation description
vacationActive: FALSE
vacationStart: 01 gennaio 2004
vacationEnd: 01 gennaio 2004
vacationForward: user@example.tld
forwardActive: FALSE
lastChange: 1108499016
cn: John
quota: 52428800S
amavisSpamKillLevel: 6.0
amavisSpamTag2Level: 5.5
amavisSpamTagLevel: 3.0
amavisBypassVirusChecks: FALSE
amavisBypassSpamChecks: TRUE
mailAutoreply: john.doe@example.tld.autoreply

dn:mail=john.smith@example.tld,vd=example.tld,o=hosting,dc=example,dc=tld
objectClass: top
objectClass: VirtualMailAccount
objectClass: Vacation
```

```
objectClass: amavisAccount
objectClass: VirtualForward
mail: john.smith@example.tld
vdHome: /home/vmail/domains
mailbox: example.tld/john.smith/
delete: FALSE
accountActive: TRUE
sn: Smith
userPassword: {SSHA}G7vRIKNRvDQg0T0qarcMgbYaQ+iOLEgq
description: Vacation description
vacationActive: FALSE
vacationStart: 01 gennaio 2004
vacationEnd: 01 gennaio 2004
vacationForward: user@example.tld
amavisBypassVirusChecks: TRUE
amavisBypassSpamChecks: FALSE
forwardActive: FALSE
lastChange: 1108499023
cn: John
quota: 52428800S
amavisSpamKillLevel: 6.0
amavisSpamTag2Level: 5.5
amavisSpamTagLevel: 3.0
mailAutoreply: john.smith@example.tld.autoreply
```

El archivo anterior es la plantilla de ejemplo, necesita algunas modificaciones para que funcione en el servidor, por ejemplo, es necesario cambiar el dominio virtual, la localización de los buzones el nombre de usuario, la contraseña, etc. Un ejemplo de una entrada para un usuario es:

```
dn: mail=nuevo1@cancun2.fi-a.unam.mx,vd=cancun2.fi-
a.unam.mx,o=hosting,dc=cancun2,dc=fi-a,dc=unam,dc=mx
objectClass: top
objectClass: VirtualMailAccount
objectClass: Vacation
objectClass: amavisAccount
objectClass: VirtualForward
mail: nuevo1@cancun2.fi-a.unam.mx
vdHome: /home/vmail/domains
mailbox: cancun2.fi-a.unam.mx/nuevo1/
delete: FALSE
accountActive: TRUE
sn: nuevo1
userPassword: {SSHA}YZQbMoKjqaj3Z1YWyRv2+8ZfeW00l1Nm
description: Vacation description
vacationActive: FALSE
vacationStart: 01 gennaio 2004
vacationEnd: 01 gennaio 2004
vacationForward: user@cancun2,dc=fi-a,dc=unam.mx
amavisBypassVirusChecks: TRUE
amavisBypassSpamChecks: FALSE
forwardActive: FALSE
```

```
lastChange: 1108499023
cn: nuevo1
quota: 52428800S
amavisSpamKillLevel: 6.0
amavisSpamTag2Level: 5.5
amavisSpamTagLevel: 3.0
```

La explicación de algunos datos de la plantilla se muestra a continuación:

dc=cancun2,dc=fi-a,dc=unam,dc=mx <==== Raíz del directorio
o=hosting <==== Organización de los dominios virtuales
cn=pahmm <==== Administrador de los dominios virtuales
vd=cancun2.fi-a.unam.mx <==== Dominio virtual
cn=postmaster <==== Administrador dominio virtual
mail=abuse@.....mx (alias to postmaster) <==== Alias de postmaster
maildrop= postmaster <==== Atributo de alias
mail=nuevo1@cancun2.fi-a.unam.mx <==== Cuenta de correo
vdHome= /home/vmail/domains <==== Ruta Base de correos
mailbox=cancun2.fi-a.unam.mx/nuevo1/ <==== ruta absoluta del buzón
mail=testmail@calcom.com.mx <==== Cuenta de correo

La entrada anterior contiene la información de un solo usuario virtual, pero se puede notar que guarda información acerca de muchas características del usuario, es por ello que es tan útil un servicio de directorio.

Ya que se tiene modificada la plantilla de los usuarios podemos dar de alta esos datos con el comando:

```
#ldapadd -v -x -D "cn=Manager,dc=cancun2,dc=fi-a,dc=unam,dc=mx" -W –h\
localhost -f /usr/local/ldap/etc/openldap/usuario.ldif
ldap_initialize( ldap://localhost )
Enter LDAP Password: PASSWORD
```

En caso de ser necesario crear muchos usuarios se puede realizar un script para automatizar la tarea. Existen muchas herramientas para migrar los usuarios del sistema a LDAP, por ejemplo, MigrationTools, scripts en perl, scripts en shell´s, etc.

Finalmente, se pueden ajustar algunos parámetros para mejorar el rendimiento de OpenLDAP, el primero de ellos es ajustar los índices, el segundo es registrar el mínimo de información es las bitácoras y por último ajustar algunas directivas en el archivo de configuración de openldap, específicamente en la sección de base de datos.

```
cachesize      70000
dbcachesize    70000
```

La directiva cachesize especifica cuántos valores de los atributos mantener en memoria. La directiva dbcachesize especifica el tamaño, en bytes, del espacio dedicado para almacenar los archivos de la base de datos en memoria. Idealmente, cachesize es tan largo como el número de entradas en tu base de datos, y dbcachesize es tan largo como todos los archivos en la base de openldap (bdb). Si no se tiene mucha memoria en el servidor es mejor mantener estos valores a un número no muy grande.

6.2.6.2 POSTFIX

Se instala openssl, para que postfix se compile con soporte de ssl (tls).
NOTA: Siempre es importante verificar las firmas de todos los paquetes que se instalan (en este escrito se omiten).

```
#tar -zxvf openssl-0.9.8h.tar.gz
#cd openssl-0.9.8h
#./config
#make
#make test
#make install
```

Crear el usuario Postfix:

```
#groupadd postdrop
#groupadd postfix
#useradd -d /no/existe -s /no/shell -g postfix -c "Postfix" postfix
```

Se procede a instalar postfix, con soporte para ldap, tls y sasl.

```
#tar -zxvf postfix-2.5.1.tar.gz
#cd postfix-2.5.1
#make makefiles CCARGS="-DUSE_TLS -DUSE_SASL_AUTH -DHAS_LDAP \
-I/usr/include/sasl2 -I/usr/local/ldap/include \
-I/usr/local/Berkeley.4.5/include" AUXLIBS="-lssl -lcrypto -lldap –llber\
-lsasl2 -L/usr/lib -L/usr/local/ldap/lib -L/usr/lib/sasl2"
#make
#make install
```

Configuración de Postfix

El archivo de configuración principal es main.cf, y se instala por defecto en /etc/postfix.

```
# Configurar el hostname
myhostname = cancun2.fi-a.unam.mx
```

```
# Configurar el nombre del dominio principal:
mydomain = cancun2.fi-a.unam.mx

# Configurar el nombre del dominio con el que salen los correos
myorigin = $mydomain

# Configurar las interfaces de red en la que escuchara peticiones
inet_interfaces = all

# Desactivar el uso del comando SMTP Vrfy
disable_vrfy_command = yes

# Dominios en los cuales recibe correo:
mydestination = localhost, $myhostname, localhost.$mydomain, $mydomain

#  alias de los correos
alias_maps = hash:/etc/postfix/aliases
#
alias_database = hash:/etc/postfix/aliases

# banner para las conexiones SMTP
smtpd_banner = $myhostname ESMTP $mail_name (Correo Ingeniería)

# Ubicación de los buzones
home_mailbox = Maildir/


# limite en tamaño de un mensaje 50MB
message_size_limit = 50000000
```

Configuración de postfix para realizar búsquedas en ldap

Se debe especificar el usuario con el que se harán las búsquedas en el directorio, el password de dicho usuario y la jerarquía en la cual iniciará la búsqueda, así como el servidor donde está implementado ldap.

```
ldap_bind_dn = cn=phamm,o=hosting,dc=cancun,dc=fi-a,dc=unam,dc=mx
ldap_bind_pw = password_phamm
ldap_search_base = o=hosting,dc=cancun2,dc=fi-a,dc=unam,dc=mx
ldap_domain = dc=cancun2,dc=fi-a,dc=unam,dc=mx
ldap_server_host = localhost
ldap_server_port = 389
ldap_version = 3
```

Debido a que el servidor soporta varios dominios virtuales, se modificará el parámetro de transportes que redirecciona los correos dependiendo del dominio virtual que estemos manejando, con esta directiva también se puede establecer la entrega de correo mediante un MDA.

```
# transports
transport_server_host = $ldap_server_host
```

```
transport_search_base = $ldap_search_base
transport_query_filter =
(&(&(vd=%s)(objectClass=VirtualDomain))(accountActive=TRUE)(delete=FALSE)
)
transport_result_attribute = postfixTransport
transport_cache = no
transport_bind = yes
transport_scope = one
transport_bind_dn = $ldap_bind_dn
transport_bind_pw = $ldap_bind_pw
```

También es posible manejar alias virtuales para especificar cuentas alternas o sinónimos de usuarios, la información puede estar contenida en el servicio de directorio, y se puede configurar en el archivo de postfix.

```
# aliases
aliases_server_host = $ldap_server_host
aliases_search_base = $ldap_search_base
aliases_query_filter =
(&(&(objectClass=VirtualMailAlias)(mail=%s))(accountActive=TRUE))
aliases_result_attribute = maildrop
aliases_bind = yes
aliases_cache = no
aliases_bind_dn = $ldap_bind_dn
aliases_bind_pw = $ldap_bind_pw
```

Un punto importante es indicarle a postfix cómo buscar la información acerca de las cuentas de los usuarios.

```
# cuentas
accounts_server_host = $ldap_server_host
accounts_search_base = $ldap_search_base
accounts_query_filter =
(&(&(objectClass=VirtualMailAccount)(mail=%s))(forwardActive=FALSE)(accou
ntActive=TRUE)(delete=FALSE))
accounts_result_attribute = mailbox
accounts_cache = no
accounts_bind = yes
accounts_bind_dn = $ldap_bind_dn
accounts_bind_pw = $ldap_bind_pw
```

Se usa el filtro para buscar el nombre del correo que tenga el valor del atributo forwardActive a FALSE y que delete sea FALSE. El resultado que buscaremos es mailbox, el cual es la ruta en el sistema de archivos del buzón de correo. Esta es la fuente LDAP mas usada porque es aquí donde residen las cuentas de correo.

Una segunda fuente LDAP para las cuentas es necesitada para ayudar a Postfix a determinar si una dirección de correo es válida antes de que

Postfix intente recibir el correo. Entonces se usa la fuente LDAP accountsmap la cual el atributo que se busca es la dirección de el correo electrónico.

```
accountsmap_server_host = $ldap_server_host
accountsmap_search_base = $ldap_search_base
accountsmap_query_filter =
(&(&(objectClass=VirtualMailAccount)(mail=%s))(forwardActive=FALSE)(accou
ntActive=TRUE)(delete=FALSE))
accountsmap_result_attribute = mail
accountsmap_cache = no
accountsmap_bind = yes
accountsmap_bind_dn = $ldap_bind_dn
accountsmap_bind_pw = $ldap_bind_pw
```

Postfix necesita saber la base donde se almacenarán los correos y el identificador del usuario del sistema que se ocupa para las cuentas virtuales.

```
# Entrega de correo
virtual_mailbox_base = /home/vmail/domains
virtual_mailbox_maps = ldap:accounts
virtual_minimum_uid = 502
virtual_uid_maps = static:502
virtual_gid_maps = static:503
```

Ya que se configuraron las búsquedas del servicio de directorio se deben actualizar las variables para que tomen el valor de acuerdo a los parámetros que se acaban de configurar.

```
mydestination = $transport_maps, localhost,
$myhostname,localhost.$mydomain, $mydomain
transport_maps = ldap:transport
virtual_maps = ldap:virtualforward, ldap:aliases, ldap:accountsmap
local_recipient_maps = proxy:unix:passwd.byname, $alias_maps,
$virtual_mailbox_maps
```

Es hora de iniciar el Postfix y comprobar que el servicio de correo funciona.

```
#/usr/sbin/postfix start
```

6.2.6.3 SERVIDOR WEB (APACHE)

Para implantar el servicio de webmail se necesita un servidor web. El servidor por excelencia hoy en día es apache, es el software más utilizado para servir páginas web en el mundo. Por lo tanto casi todo el software que se desarrolla para web tiene fácil integración con apache.

Instalación

Después de descargarse los paquetes y corroborar las firmas para evitar la corrupción en los paquetes se procede como sigue:

```
#tar -zxvf httpd-2.2.8.tar.gz
#cd httpd-2.2.8
#./configure --enable-so --enable-ssl --with-ssl=/ruta/de/openssl\
--enable-rewrite
```

Con lo anterior le especificamos, que habilite el soporte para php (se necesita para phamm), que tenga soporte de las librerías de openssl (para utilizar el protocolo https) y habilite la sobre escritura de las direcciones.

Es necesario compilar e instalar apache.

```
#make && make install
```

Creación del Certificado para el Servidor Web

Como se pretende realizar conexiones seguras mediante https, se necesita crear el certificado y firmarlo, para ello se realizó el siguiente script, que lo hace de manera automática.

```
#Script para crear el certificado y la llave
mkdir /usr/local/apache2/ssl.crt
mkdir /usr/local/apache2/ssl.key
/usr/local/bin/openssl genrsa -des3 1024 >
/usr/local/apache2/ssl.key/server.key
/usr/local/bin/openssl req -new -key
/usr/local/apache2/ssl.key/server.key > \
/usr/local/apache2/ssl.crt/server.csr
/usr/local/bin/openssl req -x509 -days 365 -key \
/usr/local/apache2/ssl.key/server.key \
-in /usr/local/apache2/ssl.crt/server.csr > \
/usr/local/apache2/ssl.crt/server.crt
cd /usr/local/apache2/ssl.key
cp server.key server.key.org
/usr/local/bin/openssl rsa -in server.key.org -out server.key
```

Al ejecutarse el script, pedirá ciertos datos para crear el certificado, como lo son el país, Ciudad, el nombre del servidor, Unidades organizativas, etc. Una vez creado el certificado el mismo script crea una llave que sirve para desencriptar el canal de comunicación una vez que la información entre al servidor, la llave se crea de manera encriptada, pero el servidor siempre pedirá la clave antes de levantar el servidor, por lo que si se desea que se haga de forma no interactiva se

debe desencriptar la llave. Esto se hace con la última línea del script. Como la llave está desencriptada es necesario guardarla con los permisos adecuados para que no pueda ser vista por nadie a excepción del administrador del servidor.

Configuración del Servidor Web

El archivo principal de configuración se encuentra en la ruta donde se instala el servidor (en caso de no especificarle una, se instala por default en /usr/local/apache2), en el directorio conf. El archivo principal de configuración es httpd.conf, en este archivo es importante deshabilitar los iconos y los manuales, ya que lo único importante es servir las páginas que el administrador desee.

```
#vi /usr/local/apache2/conf/httpd.conf

User daemon
Group daemon #el usuario y grupo debe ser diferente de root (seguridad)
ServerAdmin webadmin@example.com  (cuenta de correo en caso de problemas)
#ServerName www.example.com

#Se debe deshabilitar el directorio raíz
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

#Habilitar conexiones con transporte cifrado
# Secure (SSL/TLS) connections
Include conf/extra/httpd-ssl.conf  #ruta de archivo para https

Editar el archivo /usr/local/apache2/conf/extra/httpd-ssl.conf

#vi usr/local/apache2/conf/extra/httpd-ssl.conf

#Se indica la ruta del archivo que contiene el certificado
SSLCertificateFile "/usr/local/apache2/ssl.crt/server.crt"

#se indica la ruta que tiene la llave del certificado
SSLCertificateKeyFile "/usr/local/apache2/ssl.key/server.key"


El archivo que contiene la ruta de la llave debe ser de lectura sólo para
el administrador del sistema.

#chmod 600 "/usr/local/apache2/ssl.key/server.key

Iniciar Apache
```

```
#/usr/local/apache2/bin/apachectl start
```

6.2.6.4 Phamm (PHP LDAP Virtual Hosting Manager)

Phamm es un software de administración de LDAP para servicios virtuales, tiene plugins para correo electrónico, ftp, proxy, etc. La integración con apache y postfix es fácil de implementar. Ésta aplicación ayuda a la administración del servicio de LDAP para ciertos servicios, está programado en php y no consume muchos recursos del sistema.

Debido a que phamm está programado en php, es necesario que se instale este paquete

```
#tar -zxvf  php-version.tar.gz
#cd php-version
#./configure –prefix=/usr/local/php –with-config-file-\
path=/usr/local/php\
 --with-apxs2=/usr/local/apache2/bin/apxs –-with-mysql –-with-openssl
#make && make install
```

Para que apache pueda interpretar código php, añadimos las siguientes líneas al archivo de configuración de apache.

```
#vi  /usr/local/apache2/conf/httpd.conf

LoadModulecphp5_module   modules/libphp5.so
# If the AddEncoding directives above are commented-out, then you
# probably should define those extensions to indicate media types:

AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType application/x-httpd-php .php
```

Instalación de Phamm

Phamm se descomprimió cuando se hizo la configuración de OpenLdap con postfix(para usar los schemas en postfix), por lo que sólo resta copiar los archivos de phamm al directorio que mostrará nuestro servidor web.

```
# cp -r -v phamm-0.5.10 /var/www/localhost/htdocs/phamm
# cd /var/www/localhost/htdocs/phamm
```

Configuración de Phamm

El archivo global de configuración de Phamm es:

```
/var/www/localhost/htdocs/phamm/config.inc.php

# vim /var/www/localhost/htdocs/phamm/config.inc.php

// The server address (IP or FQDN)
define ('LDAP_HOST_NAME','127.0.0.1');

// The protocol version [2,3]
define ('LDAP_PROTOCOL_VERSION','3');

// The server port
define ('LDAP_PORT','389');

// The container
define ('SUFFIX','dc=cancun,dc=fi-a,dc=unam,dc=mx');

// The admin bind dn (could be rootdn)
define ('BINDDN','cn=Manager,dc=cancun,dc=fi-a,dc=unam,dc=mx');

// The Phamm container
define ('LDAP_BASE','o=hosting,dc=cancun,dc=fi-a,dc=unam,dc=mx');
 # Cambiar FORCE_SSL a 1 para obligar el acceso vía SSL.
define ('FORCE_SSL',1);

// *============================*
// *===   Plugins Settings   ===*
// *============================*

// The default plugin
define ('DEFAULT_PLUGIN','mail');


# Ya que usamos el hash md5 para las contraseñas de los usuarios en
#OpenLDAP,
# también usaremos este algoritmo en Phamm:
define ('ENC_TYPE','md5');    // Standard LDAP encryption type
```

El archivo de configuración de plugin mail es:

```
/var/www/localhost/htdocs/phamm/plugins/mail.xml
```

En el archivo que contiene el plugin de correo se pueden modificar parámetros generales de las cuentas de usuario, y tener una plantilla con opciones por default.
Una vez configurado phamm, es necesario indicarle al servidor web el directorio donde reside phamm y otorgarle permisos sólo a ciertos equipos. Para hacerlo, se edita el archivo de configuración de apache.

```
#vi /usr/local/apache2/conf/httpd.conf

#Agregar en la sección de alias
Alias /administracion /var/www/localhost/htocs/phamm
```

```
<Directory /var/www/localhost/htdocs/phamm>
Options SymLinksIfOwnerMatch IncludesNoExec
AllowOverride None
Order Deny,Allow
Deny from All
Allow from 127.0.0.1 #Se escriben las IP que administrarán el servicio
<Directory /var/www/localhost/htdocs/phamm>
```

Es necesario reinciar apache, para que surtan efecto los cambios.
#/usr/local/apache2/bin/apachectl restart

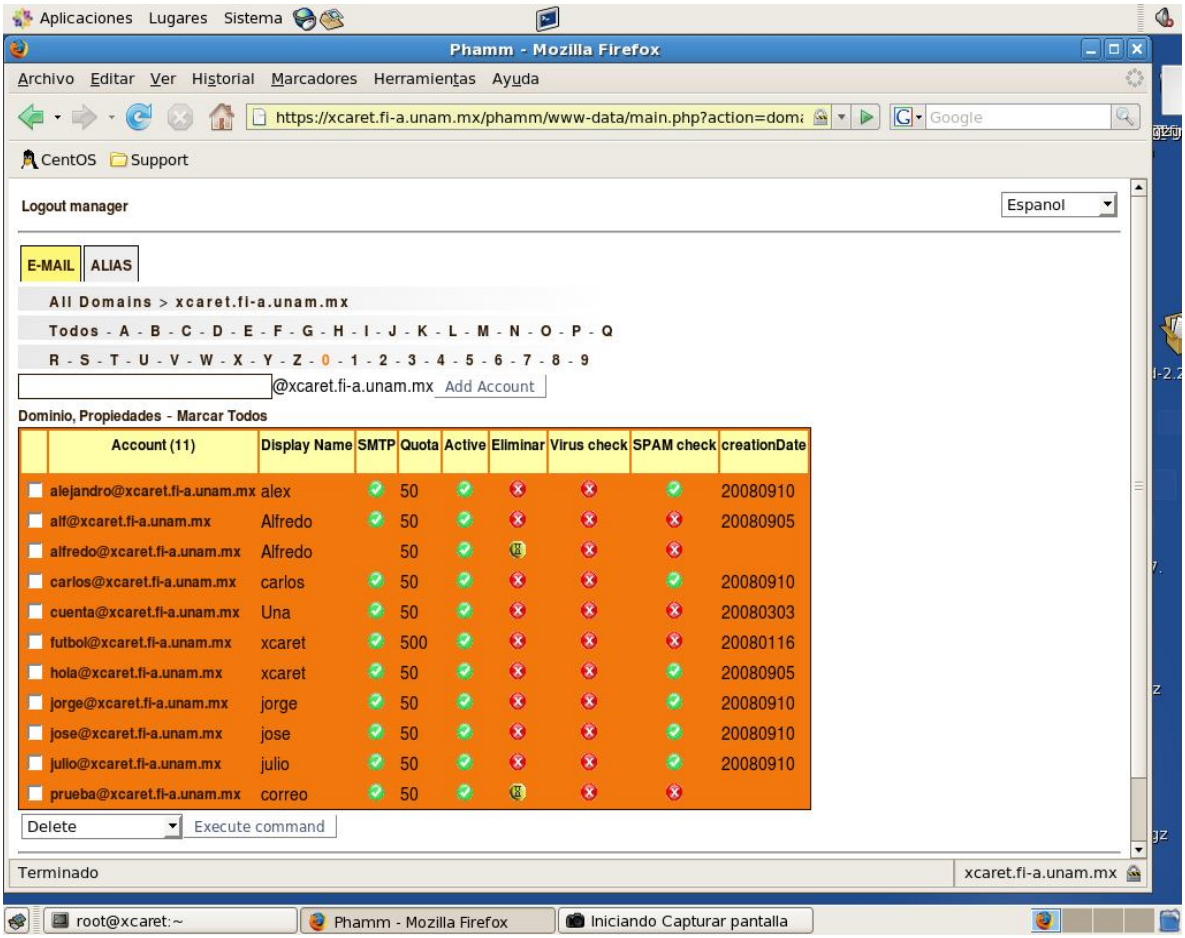


Figura 6.3 Pantalla de usuarios de correo con Phamm

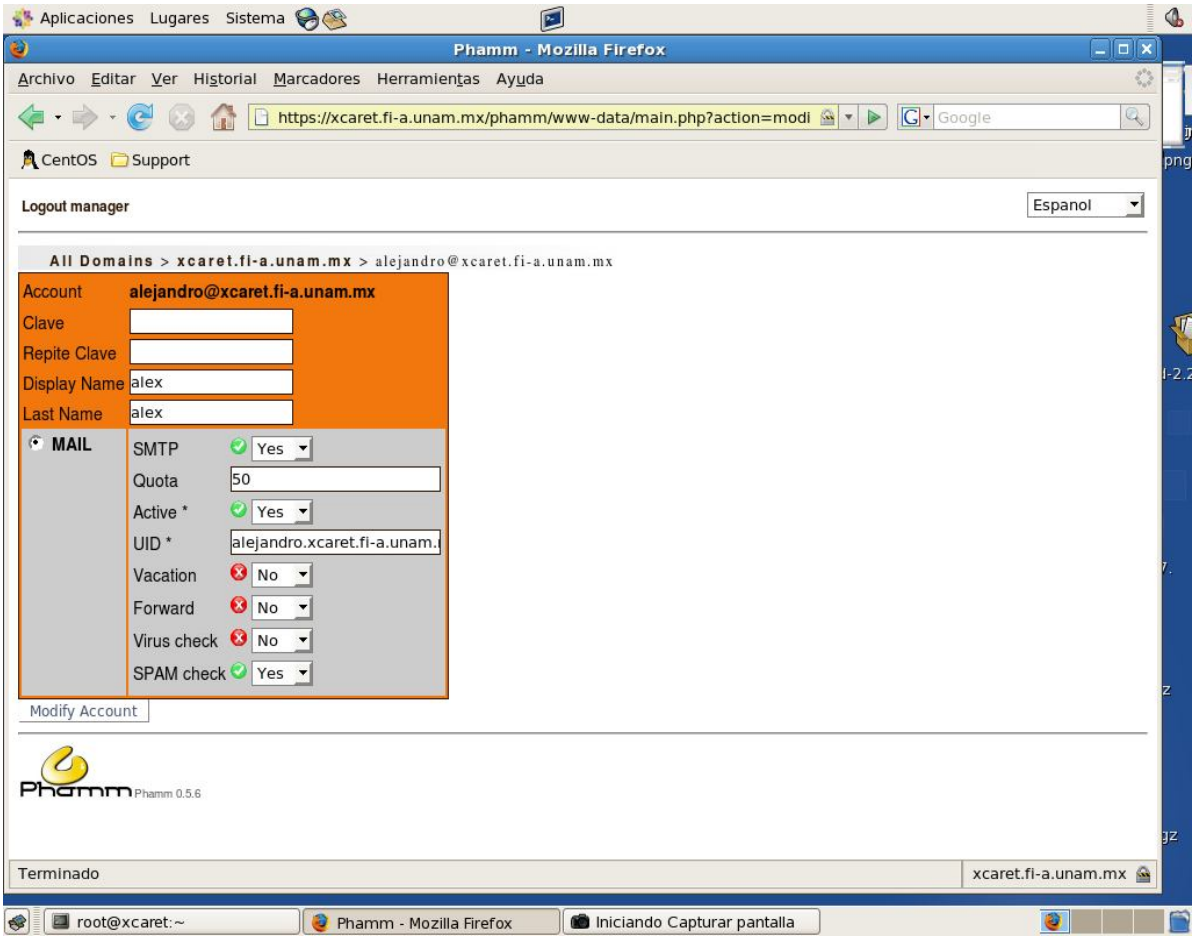


Figura 6.4 Edición de cuentas de correo con Phamm

6.2.6.5 Dovecot

Los servicios de POP3 e IMAP los proporcionará dovecot.

Instalación

```
#tar –zxvf dovecot-version.tar.gz
#cd dovecot.version
#./configure --with-ldap -with-ssldir=/ruta/instalacion/ssl
#make
#make install

Se crea el usuario dovecot

# groupadd  dovecot
# useradd –g dovecot –d /usr/libexec/dovecot dovecot
```

La configuración de dovecot se realiza en el archivo /etc/dovecot.conf

```
#vi /etc/dovecot.conf
login_user = dovecot #dovecot-auth
protocols = imap pop3
ssl_disable = yes
disable_plaintext_auth = no
#client_workarounds = oe6-fetch-no-newmail outlook-idle
log_timestamp = "%Y-%m-%d %H:%M:%S "
#mail_extra_groups = mail
mail_debug = no
mail_location = maildir:/home/vmail/domains/cancun2.fi-
a.unam.mx/%u/Maildir
#mail_location = maildir:/home/vmail/domains/%d/%u
auth default {
    mechanisms = plain
    passdb ldap {
        args = /etc/dovecot.phamm-ldap.conf #password en ldap
    }
    userdb ldap {
        args = /etc/dovecot.phamm-ldap.conf #usuarios en ldap
    }
}
#dict {
#}
#plugin {
#}
auth_verbose = yes
auth_debug = yes
```

En el archivo anterior se especificó que los usuarios y las contraseñas se almacenan en el archivo `/etc/dovecot.phamm-ldap.conf`

```
#vi /etc/dovecot.phamm-ldap.conf
```

```
hosts = localhost
#auth_bind = yes
auth_bind = no
dn = cn=Manager,dc=cancun2,dc=fi-a,dc=unam,dc=mx
dnpass=password
#auth_bind_userdn = mail=%u,vd=%d,o=hosting,dc=cancun2,dc=fi-
a,dc=unam,dc=mx  ----checar el %d
#auth_bind_userdn = mail=%u@cancun2.fi-a.unam.mx,vd=cancun2.fi-
a.unam.mx,o=hosting,dc=cancun2,dc=fi-a,dc=unam,dc=mx
ldap_version = 3
base = o=hosting,dc=cancun2,dc=fi-a,dc=unam,dc=mx
deref = never
scope = subtree
user_attrs = mail
#user_attrs = uidNumber=502,gidNumber=503
user_filter = (mail=%u@cancun2.fi-a.unam.mx)
#user_filter =
(&(objectClass=VirtualMailAccount)(accountActive=TRUE)(mail=%u@cancun2.fi
-a.unam.mx))
#pass_attrs = uid=cn,userPassword=password
pass_filter =
(&(objectClass=VirtualMailAccount)(accountActive=TRUE)(mail=%u@cancun2.fi
-a.unam.mx))
default_pass_scheme = MD5
# the uid of your vmail user
user_global_uid = 502
# the guid of your vmail group
user_global_gid = 503
```

Si se pretende que dovecot intercambie información sobre un canal seguro es necesario indicarle la ruta del certificado y de la llave. Podemos ocupar el mismo certificado que se creó para apache. Las modificaciones en el archivo de configuración son las siguientes.

```
#vi /etc/dovecot.conf

protocols = imaps pop3s
ssl_disable = no
ssl_cert_file = /ruta/al/certificado
ssl_key_file = /ruta/a/lallave
```

Iniciar dovecot (se puede crear un script con el ejecutable, para que inicie el servicio al iniciar el servidor)

```
#dovecot
```

SMTP Autenticado (SMTP-AUTH)

Con SMTP-AUTH se configura el MTA (Postfix) para que cuando se establezca una conexión con el servidor el usuario primero se autentique

con su usuario (email) y contraseña antes de que pueda enviar correos, si el usuario se autentica satisfactoriamente, es decir, el usuario y la contraseña son válidas entonces el usuario podrá enviar correo por medio de el servidor. Este procedimiento es útil para controlar el relay a cualquier máquina.

Para configurar SMTP-AUTH en Postfix se usara SASL que proporciona dovecot. Para saber si postfix tiene soporte para dovecot SASL, escribir:

#postconf –a

Una vez comprobado que se tiene el soporte para SASL, se debe modificar el archivo de configuración de postfix para indicarle el método de autenticación.

Configurar el mecanismo de autenticación.

```
# vi /etc/postfix/main.cf

smtpd_sasl_auth_enable = yes
smtpd_recipient_restrictions =
permit_mynetworks,permit_sasl_authenticated,reject_unauth_destination
smtpd_sasl_security_options = noanonymous
smtpd_sasl_authenticated_header = yes
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth

Ahora es necesario configurar dovecot

# vi /etc/dovecot.conf

login_chroot = yes
login_user = postfix
first_valid_uid = 591 #  UID para "postfix" de /etc/passwd
pop3_uidl_format = %08Xu%08Xv
auth default {
    mechanisms = PLAIN LOGIN
    passdb ldap {
        args = /etc/dovecot-ldap.conf
    }
    userdb ldap {
        args = /etc/dovecot-ldap.conf
    }
socket listen {
                master {
                        path = /var/run/dovecot/auth-master
                                mode = 0600
                        user = vmail
                        group = vmail
                }
                client {
```

```
                        path = /var/spool/postfix/private/auth
                        mode = 0660
                        user = postfix
                        group = postfix
                }
        }
        user = vmail
}
```

Reiniciar (también dovecot)postfix para que los cambios surtan efecto:

```
# /etc/init.d/postfix restart
```

La figura que se muestra a continuación tiene configurados los parámetros para que el flujo de información viaje por un canal cifrado.

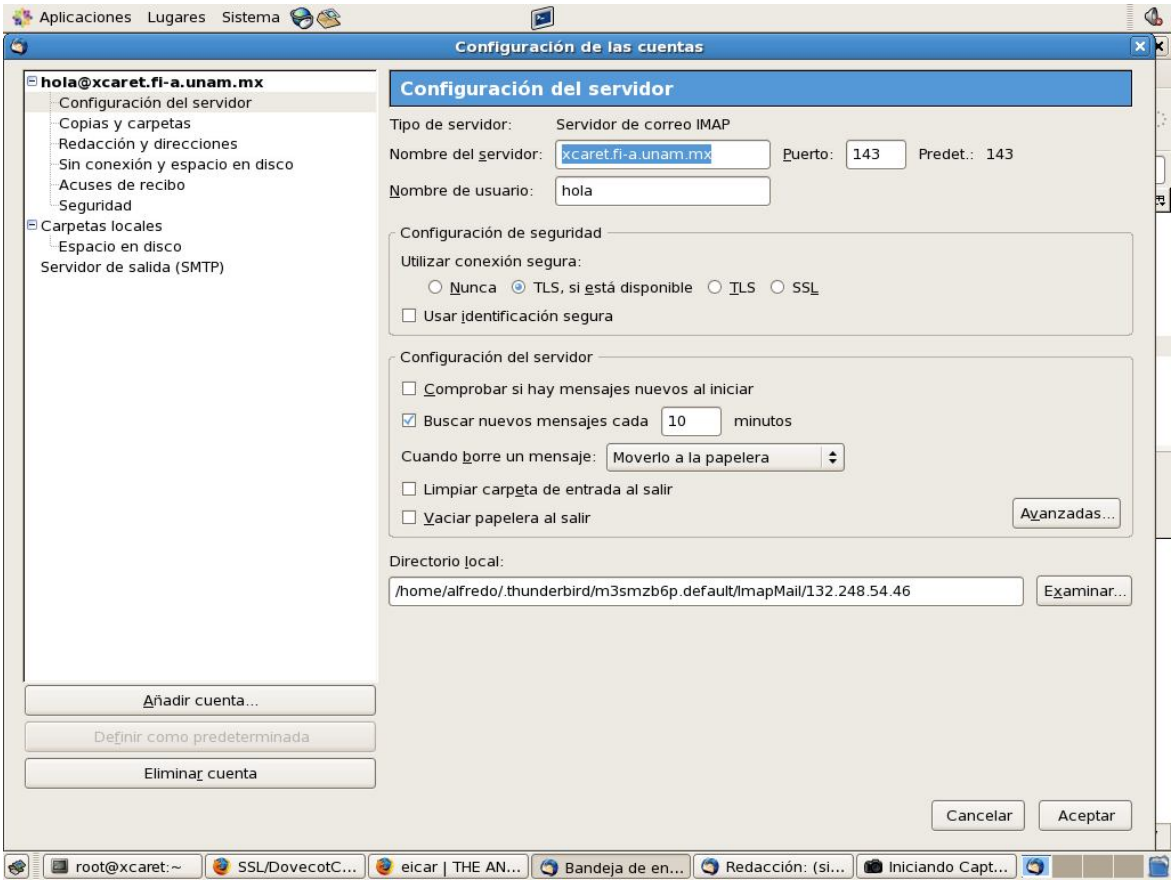


Figura 6.5 Configuración de Thunderbird (MUA), para el soporte de tls

Ahora que se ha configurado dovecot para transferir información sobre ssl, también es posible configurar postfix para que los mensajes viajen por un canal seguro, las únicas modificaciones que se tienen que hacer al archivo de configuración de postfix son:

#vi /etc/postfix/main.cf

```
smtpd_use_tls = yes
smtpd_tls_key_file = /ruta/llave
smtpd_tls_cert_file = /ruta/certificado
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
```

Al reiniciar postfix se debe tener un servicio de smtp sobre ssl.

6.2.6.6 MailScanner (ANTIVIRUS Y ANTISPAM)

Hasta el momento se tiene configurado los servicios de SMTP, IMAP y POP3 sobre un canal seguro, pero una vez que lleguen los mensajes al servidor (o salgan de él) será necesaria la revisión de dichos mensajes mediante un scanner, el cual buscará virus o mensajes con spam. El scanner que se instalará tendrá a f-prot como antivirus y a Spamassassin como antispam.

La instalación de f-prot y spamassassin es trivial, por lo que esta sección se centrará en la configuración del scanner.

```
tar xvzf Mailscanner-<version>.rpm.tar.gz
cd Mailscanner-<version>
export LANG=C; ./install.sh
```

Detenemos el servicio postfix:

```
#/usr/sbin/postfix stop
```

Se activa el inicio de mailscanner de la siguiente manera:

```
#chkconfig MailScanner on
```

Editar el archivo /etc/postfix.in/main.cf y colocamos el siguiente texto en el inicio del archivo

```
header_checks = regexp:/etc/postfix/header_checks
/^Received:/ HOLD
```

Editar el archivo de configuración de Mailscanner

```
%org-name% = servidor
Run As User = postfix
Run As Group = postfix
```

```
Incoming Queue Dir = /var/spool/postfix/hold
Outgoing Queue Dir = /var/spool/postfix/incoming
MTA = postfix

File Timeout = 120
Maximum Archive Depth = 20
Virus Scanners = f-prot

Use SpamAssassin = yes
SpamAssassin User State Dir = /var/spool/MailScanner/spamassassin


Spam List = ORDB-RBL SBL+XBL SORBS-DNSBL CBL RSL DSBL spamcop
Minimum Stars If On Spam List = 3
Spam Lists To Be Spam = 2
Sign Messages Already Processed = no
Sign Clean Messages = no

MTA=postfix

Use SpamAssassin = yes

Required SpamAssassin Store
High SpamAssassin Score = valor
```

Ahora necesitamos asegurarnos de la que el usuario postfix pueda escribir en los siguientes directorios, para lo cual ejecutamos los siguientes comandos

```
chown -R postfix.postfix /var/spool/MailScanner/incoming
chown -R postfix.postfix /var/spool/MailScanner/quarantine

mkdir -m 700 /var/spool/MailScanner/spamassassin
chown -R postfix.postfix /var/spool/MailScanner/spamassassin
```

Se Inicia el scanner (postfix se activará de forma automática)

```
#service MailScanner start
```

6.2.6.7 Roundcube WebMail

Muchas veces los usuarios necesitan consultar su correo desde diferentes máquinas, para ello se implementa un webmail que proporciona la facilidad de recuperar los mensajes desde cualquier computadora que tenga un browser y conexión a la red. El software seleccionado para implantar el servicio de webmail es roundcube debido a sus características, por ejemplo la conexión a los correos se hace a través de IMAP, por lo que no importa el formato en el que están escrito

los mensajes, permite administración de carpetas, libreta de direcciones, y soporta MIME.

Para instalar roundcube se necesita tener instalado módulo php (con soporte mysql, postgresql o SQLite) para apache, también se necesita una base de datos para guardar información acerca de los usuarios. Soporta Postgresql, Mysql y SQLite. En la instalación se manejará con mysql. La base de datos se puede instalar con yum:

#yum install mysqld-server

Para instalar roundcube aplicar

```
#tar -zxvf roundcubemail-version.tar.gz
```

Los archivos desempaquetados se deben poner en un directorio que pueda ser servido por apache.

En Mysql se necesita una base para el webmail y un usuario.

```
#mysql
```

Una vez adentro de mysql, teclear (el usuario y el password es de acuerdo a nuestro gusto)

```
CREATE DATABASE roundcubemail;
GRANT ALL PRIVILEGES ON roundcubemail.* TO username@localhost IDENTIFIED
BY 'password';
FLUSH PRIVILEGES;
```

Se puede hacer una instalación totalmente guiada a través del instalador de roundcube, para ello se debe contar con un browser y teclear la dirección del directorio donde colocamos el directorio de roundcube

```
https://mimaquina/rouncube/installer
```

El instalador checará las dependencias y guiará en la instalación comprobando en cada una de las fases los requerimientos.

Al final del proceso de instalación se crearán dos archivos de configuración los cuales hay que copiar al directorio “config” de roundcube, dichos archivos son: main.inc.php y db.inc.php, se deben editar para adaptarlos a nuestro servidor de mail.

Nota: El usuario con que inicia apache debe tener permiso de escritura en el directorio “logs” y “temp” de roundcube.

6.2.6.8 ALTA DISPONIBILIDAD

Como ya se mencionó en el capítulo dedicado a la alta disponibilidad, heartbeat es un una solución de alta disponibilidad a través de replicación de software revisando la disponibilidad de un sistema y sustituyéndolo en caso de que ocurra un problema. Claro está, se necesitan tener dos nodos (o más) con la misma información para que heartbeat revise al nodo principal y en caso de fallo se active el nodo secundario.

Configuración

La instalación de heartbeat no representa mayor problema, casi todas las distribuciones contienen paquetes para hacerlo mediante el sistema de empaquetado, así que se procederá a la configuración del software.

Copiar en la ruta /etc/ha.d/ las plantillas de los archivos de configuración que se ubican en la siguiente ruta: /usr/share/doc/heartbeat-version

```
#cp /usr/share/doc/heartbeat-version/haresources /etc/ha.d
#cp /usr/share/doc/heartbeat-version/authkeys  /etc/ha.d
#cp /usr/share/doc/heartbeat-version/ha.cf /etc/ha.d/
#cp /usr/share/doc/heartbeat-2/ha_logd.cf /etc/logd.cf
```

En el archivo /etc/logd.cf
Descomentamos la línea logfacility y se modifica de esta forma:

```
logfacility     daemon
```

Definir las características del servicio de HeartBeat en /etc/ha.d/ha.cf
Este archivo le dice a HeartBeat qué tipos de interfaces a utilizar para comunicarse con los nodos del clúster. También define los nodos que van a formar el cluster y los archivos de log.

```
#vi /etc/ha.d/ha.cf

#Archivos de log de HB
debugfile /var/log/HA-LOG/ha-debug
logfile /var/log/HA-LOG/ha-log
keepalive 2
```

```
# tiempo en segundos en el que se considera un sistema como 'muerto' si
no responde
deadtime 30

# puerto de comunicación por la interfaz de red
udpport 694

# interfaz de red para el heartbeat
bcast ethx


# definición de nodos del cluster
node cancun2.fi-a.unam.mx
node xcaret.fi-a.unam.mx

#Habilitamos el CRM
crm yes
```

En el archivo “authkeys” se define el nivel de seguridad con la que se intercambian información los nodos del cluster. Existen tres tipos: crc, md5 y sha1. Siendo el más seguro el sha1 y el más inseguro el crc.

```
#vi /etc/ha.d/autkeys

auth
2 sha1 password
```

password es la palabra que tenemos que definir en ambos nodos para que sirva de clave entre ellos.
Una vez definida se debe cambiar los permisos de este fichero para que sólo pueda ser leído por el administrador

```
#chmod 600 /etc/ha.d/authkeys
```

Configuración de los recursos

En el archivo “/var/lib/heartbeat/crm/cib.xml” se definen los recursos que son gestionados por HeartBeat. Los recursos son scripts Linux Standard Base (LSB) como los que se usan para arrancar o parar servicios al arrancar el sistema en los diferentes runlevels. Heartbeat buscará estos scripts en estas dos rutas: /etc/rc.d/init.d y /etc/ha.d/resource.d. La secuencia que sigue HeartBeat a la hora de levantar el servicio es la siguiente: Primero comprueba que el nodo donde definimos que se debe arrancar el servicio (Nodo1) está operativo, una vez comprobado, levanta un alias con la dirección del servicio que le indiquemos en el servidor activo. Si hubiera más de un servicio, heartbeat los arrancaría de forma secuencial, esto es, de izquierda a derecha.

Editamos el archivo /etc/ha.d/haresources.bak

```
#vi /etc/ha.d/haresources

cancun2.fi-a.unam.mx IPaddr::132.248.54.242/24/eth0
```

Con lo anterior se indica que el nodo principal será “cancún2” la ip es “132.248.54.242”, la máscara de red es de 24 bits y la interfaz es eth0. De esta forma si llega a fallar el nodo principal, entonces heartbeat se encargará de cambiar la ip al nodo secundario para que los servicios se sigan ejecutando.
Es necesario crear el archivo xml, lo que se hace con el siguiente comando.

```
#/usr/lib/heartbeat/haresources2cib.py /etc/ha.d/haresources.bak
```

Esto nos crea el archivo de configuración del CRM en /var/lib/heartbeat/crm/cib.xml

Ahora cambiaremos el usuario y grupos propietarios de los directorios de HeartBeat:

```
chown -R hacluster:haclient /var/run/heartbeat
chown -R hacluster:haclient /var/lib/heartbeat
chown -R hacluster:haclient /usr/lib/heartbeat
```

A parte de los ficheros de configuración de HeartBeat hay que definir en el archivo hosts de todos los servidores del cluster las ip series y los nombres tanto de los nodos del cluster como de los alias del servicio.

```
ipnodo1       nombrenodo1  nombrenodo1.dominio
ipnodo2       nombrenodo2  nombrenodo2.dominio
ipservicio    nombreservicio  nombreservicio.dominio
```

Para el correcto funcionamiento de HeartBeat se recomienda que todos los nodos del cluster tengan la fecha y hora sincronizadas.

Todos los pasos anteriores se deben realizar también en el otro nodo del cluster. Los dos nodos deben de tener la misma información en los archivos configurados anteriormente, por lo que una recomendación es copiarlos.

Arrancar HeartBeat:

Para arrancar HB es necesario ejecutar el siguiente script con root:

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

Para la prueba de funcionamiento, se utiliza la utilidad de monitoreo crm_mon.

```
# crm_mon -i2

Last updated: Thu Aug 20 20:32:10 2008
Current DC: xcaret.fi-a.unam.mx (862ac8fb-2ca3-46c0-a511-22488998384d)
2 Nodes configured.
1 Resources configured.
============

Node: xcaret.fi-a.unam.mx (862ac8fb-2ca3-46c0-a511-22488998384d): online
Node: cancun2.fi-a.unam.mx (7f31cdfb-4a0f-429f-93f8-cecde4c4bf8b): online
```

Detener el nodo principal (cancun2) y verificar que el nodo secundario se activa.
Una vez detenido el nodo principal, se comprueba el funcionamiento con

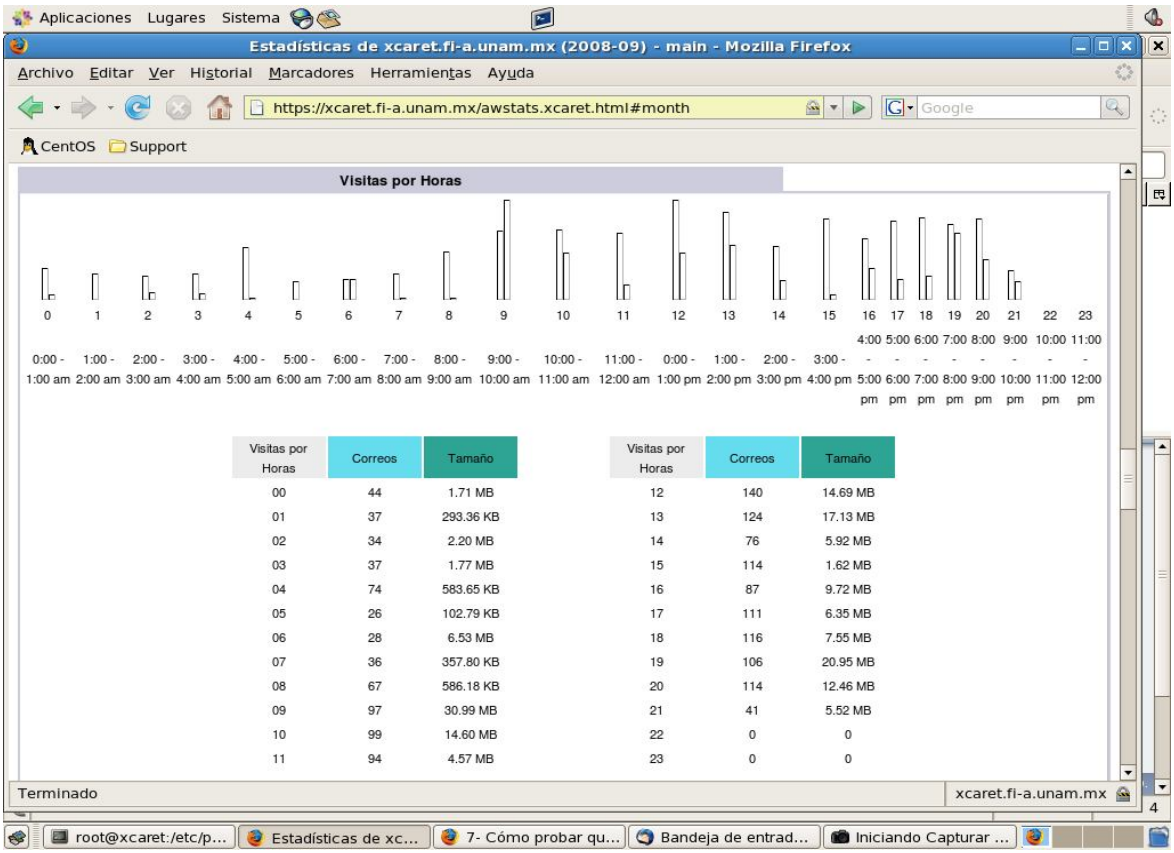
```
#crm_mon

Last updated: Thu Aug 20 20:38:15 2008
Current DC: xcaret.fi-a.unam.mx (862ac8fb-2ca3-46c0-a511-22488998384d)
2 Nodes configured.
1 Resources configured.
============

Node: xcaret.fi-a.unam.mx (862ac8fb-2ca3-46c0-a511-22488998384d): online
Node: cancun2.fi-a.unam.mx (7f31cdfb-4a0f-429f-93f8-cecde4c4bf8b):
OFFLINE

IPaddr_132_248_54_242   (ocf::heartbeat:IPaddr):        Started
xcaret.fi-a.unam.mx
```

Lo anterior muestra que el nodo secundario se activó debido a un fallo en el nodo principal, y se le asigno la ip del servidor primario.

6.2.6.9 ESTADÍSTICAS DE CORREO

Siempre es bueno contar con información acerca de la utilización del correo, porque a partir de ahí se puede tomar decisiones acerca de los límites del sistema que deben establecerse a cada uno de los servicios. Se instaló el sistema de generación de estadísticas “awstats” en una máquina alterna para que las bitácoras estén fuera del servicio principal y no se añada carga extra al servidor. Awstats es un software que permite generar estadísticas detalladas de uso del servidor de correo, así como del servidor web. Dichas estadísticas las genera con gráficas y pueden ser revisadas a través de un servidor web.

La revisión periódica de las estadísticas también reporta anomalías en el servicio, si bien es cierto que al no existir usuarios “reales” en el servidor es difícil que se comprometa la máquina a nivel de sistema, si es posible que una cuenta de correo caiga en manos no deseadas y se este cometiendo abusos a partir de dicha cuenta. En las siguientes imágenes se muestran estadísticas del sistema de correo.



Visitas por Horas	Correos	Tamaño
00	44	1.71 MB
01	37	293.36 KB
02	34	2.20 MB
03	37	1.77 MB
04	74	583.65 KB
05	26	102.79 KB
06	28	6.53 MB
07	36	357.80 KB
08	67	586.18 KB
09	97	30.99 MB
10	99	14.60 MB
11	94	4.57 MB

Visitas por Horas	Correos	Tamaño
12	140	14.69 MB
13	124	17.13 MB
14	76	5.92 MB
15	114	1.62 MB
16	87	9.72 MB
17	111	6.35 MB
18	116	7.55 MB
19	106	20.95 MB
20	114	12.46 MB
21	41	5.52 MB
22	0	0
23	0	0

Figura 6.6 Estadísticas de correos por hora (Servidor Cancún)

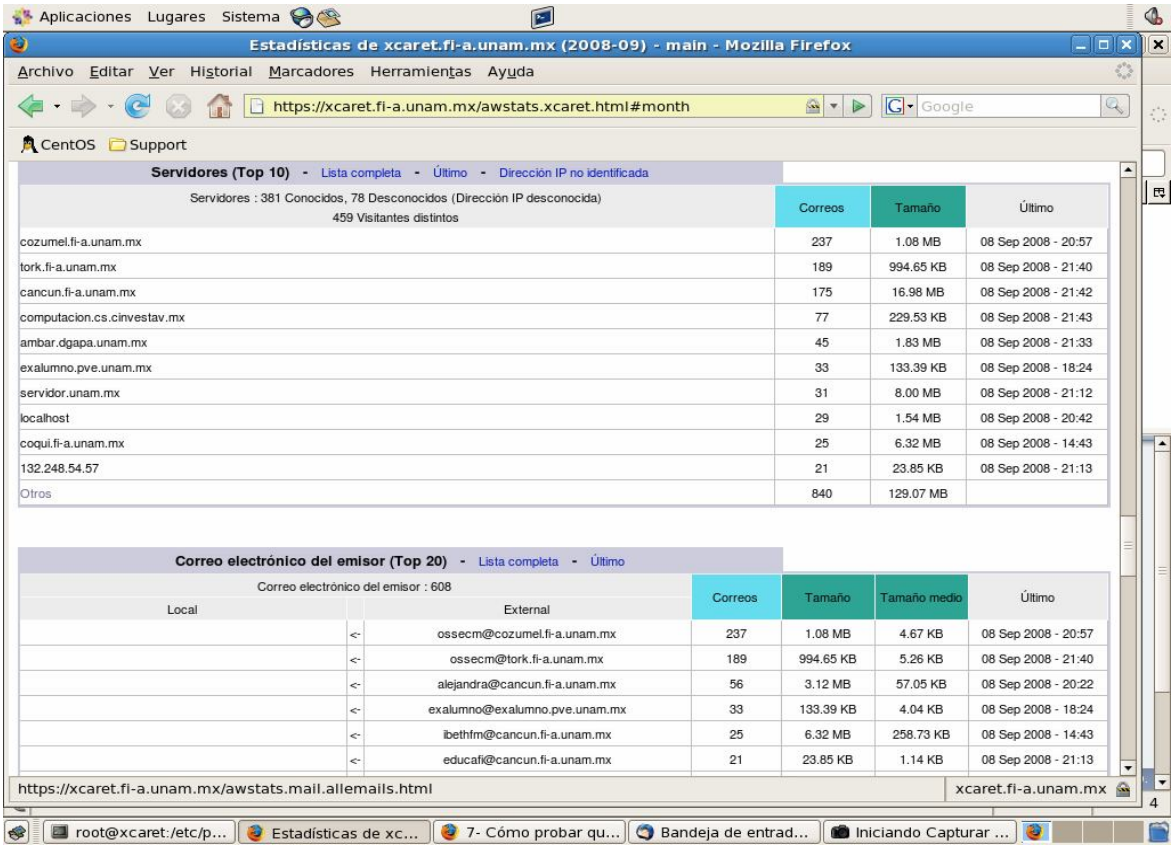


Figura 6.6 Estadísticas correo (cuentas que usan frecuentemente el servidor)

6.3 Administración

Si bien es cierto que los paquetes instalados proporcionan los servicios necesarios para el servicio de correo electrónico, quizás el componente principal en el correcto funcionamiento del sistema sean los administradores.
Se debe contar con personal capacitado que conozca a fondo cada uno de los componentes del sistema y que lleven a cabo tareas para mantener de forma permanente el servicio.

Funciones

Algunas de las funciones que deben cumplir los administradores son las siguientes:

- Revisar de forma periódica las bitácoras generadas.
- Realizar respaldos del sistema en forma constante, tener un calendario para respaldos incrementales y completos
- Realizar monitoreo periódico de los servicios del sistema así como del rendimiento de los mismos para evitar caídas en el rendimiento
- Estar informado sobre las vulnerabilidades de los paquetes instalados y tener la capacidad de aplicar parches o instalaciones.
- Realizar de forma frecuente análisis del sistema operativo y demás aplicaciones en busca de vulnerabilidades
- Memorizar las contraseñas
- Capacitarse constantemente
- Investigar nuevas herramientas para mejorar el sistema actual
- Realización de scripts que automaticen ciertas tareas
- Actualización del sistema
- Revisar el correcto funcionamiento del hardware donde reside el servicio
- Desarrollar un ambiente de pruebas

6.4 Revisión del Diseño

El resumen de los elementos que conforman el sistema es el siguiente:

Elementos de Seguridad

- Detector de Intrusos
- Hardening del sistema

- Servicios bajo canales seguros (HTTPS,POP3S,IMAPS,SMTP sobre SSL)
- Instalación de scanner de antivirus y antispam
- Estadísticas (Figura 6.6 y 6.7)
- Usuarios virtuales (Figura 6.4)
- Políticas de uso (pueden encontrarse en www.correo.fi-a.unam.mx)

Alta Disponibilidad

- Se tienen planes de respaldo, contingencia y recuperación de desastres
- Se implementó Heartbeat

Servicios

- Correo (Transferencia y recuperación), Servicio de Directorio para autenticación (Figura 6.7)
- SSH
- Webmail (Figura 6.8)
- Administración gráfica (Figura 6.9)

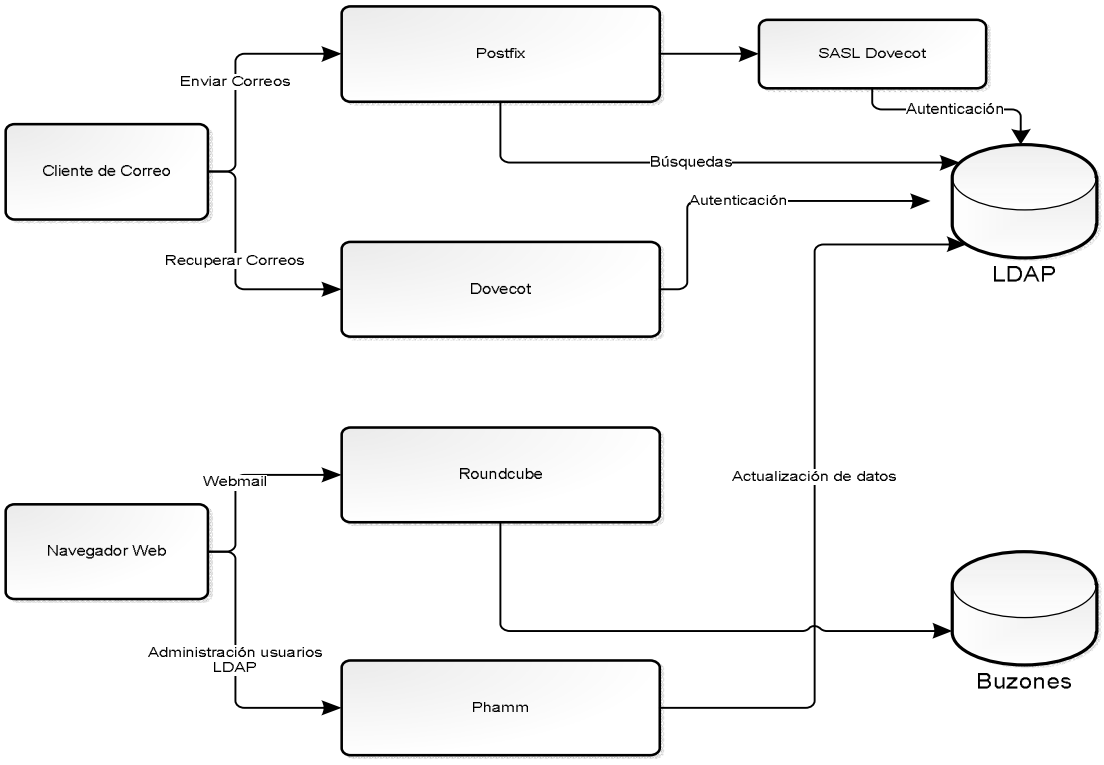


Figura 6.7 Transferencia y recuperación de correos

Todos los servicios implantados están en un servidor piloto, para que en poco tiempo sean implementados en el servidor oficial, a partir de este momento empieza la etapa de capacitación de personal, el ambiente de pruebas, los planes de administración y finalmente la puesta en marcha del servicio.

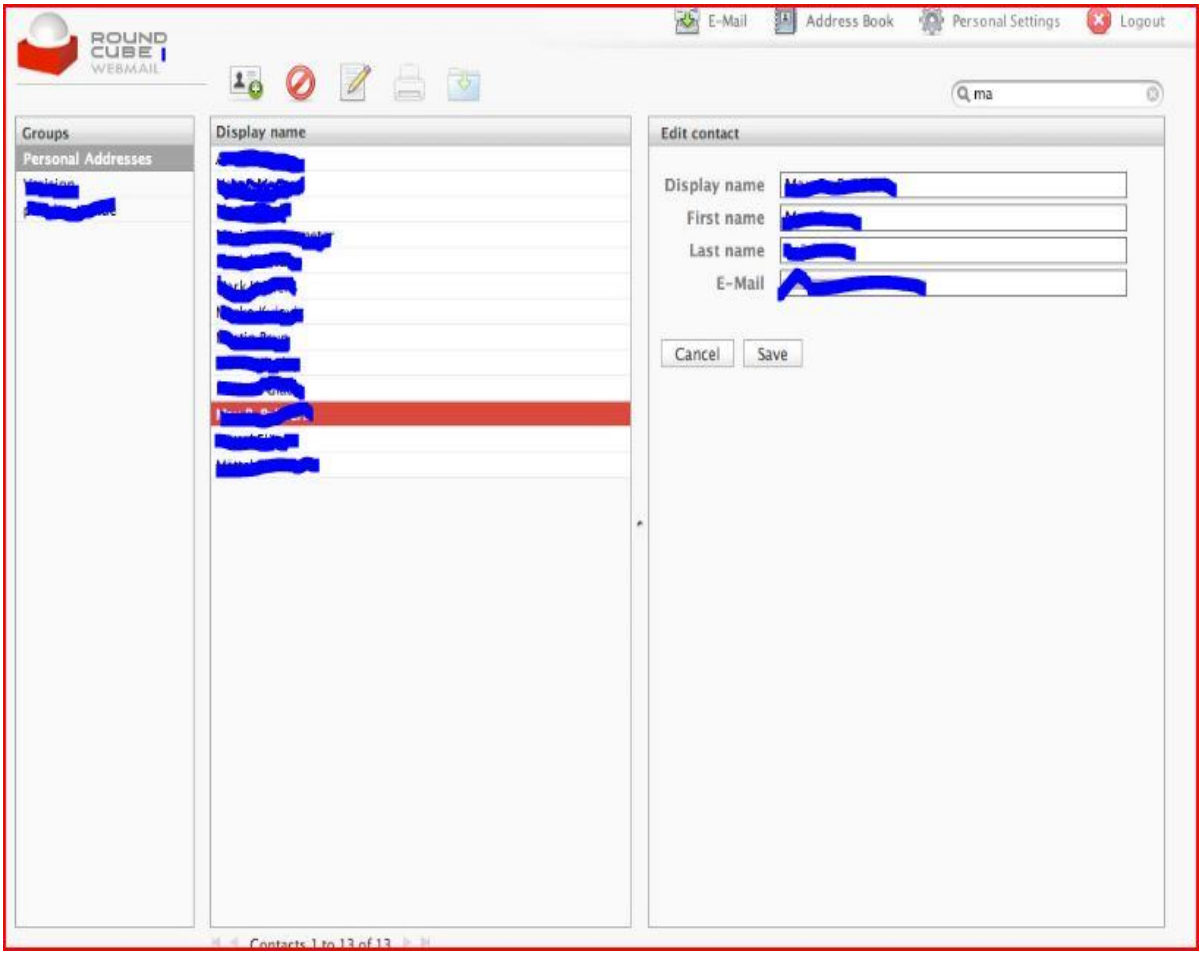


Figura 6.8 Pantalla de libreta de direcciones del Webmail

Pruebas del Servidor de Correo

Con cada uno de los servicios implementados se realizaron pruebas de funcionamiento, estos servicios se pueden comprobar del lado del cliente y la salida depende del software que utilice el usuario. Algunas pruebas que se pueden realizar en el servidor se muestran en las secciones de instalación de los paquetes, por ejemplo se tomaron imágenes de phamm, de pruebas de imap a través de thunderbird, de pruebas de heartbeat con crm_mon, etc.
A continuación se muestra una salida en texto del comando nmap para mostrar los servicios que están funcionando en el servidor local.

```
[root@xcaret ~]# nmap localhot

Starting Nmap 4.11 ( http://www.insecure.org/nmap/ ) at 2008-09-10 16:11
CDT
Interesting ports on localhost.localdomain (127.0.0.1):
Not shown: 1673 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
443/tcp open  https
993/tcp open  imaps
995/tcp open  pop3s
389/tcp open  ldap
3306/tcp open mysql

Nmap finished: 1 IP address (1 host up) scanned in 0.121 seconds
```

En la salida del comando anterior se muestra que todos los servicios propuestos están funcionando.

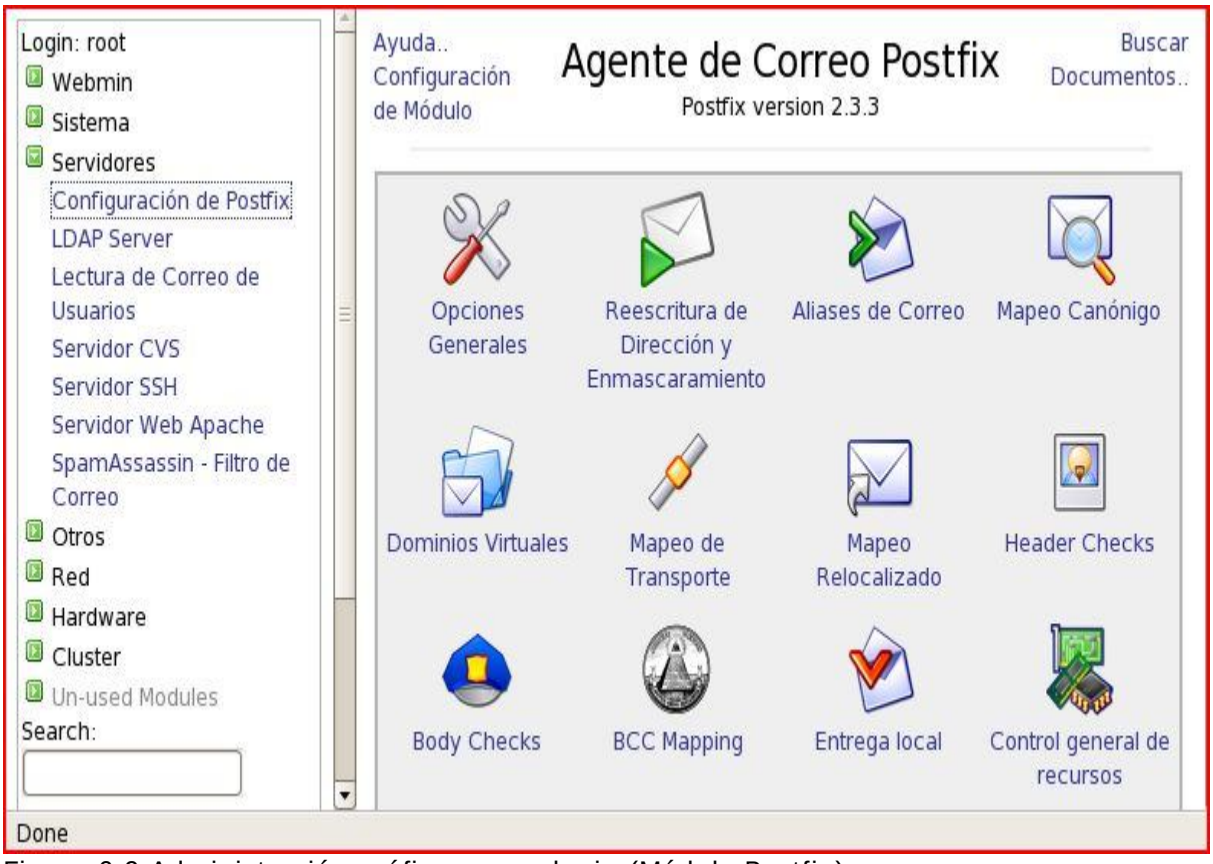


Figura 6.9 Administración gráfica con webmin (Módulo Postfix)

Aunque se comprobó el funcionamiento de cada uno de los servicios que ofrece el sistema, se programó un semestre de pruebas y capacitación al personal para implementarlo en el servidor real.
Al momento, el servidor piloto así como todos sus servicios lo está probando el departamento de Redes y Operación de Servidores (DROS), en una segunda fase se pretende que intervengan todos los elementos de UNICA para finalmente liberarlo a la comunidad de la Facultad de Ingeniería.

La siguiente figura muestra la implementación final del servidor piloto.

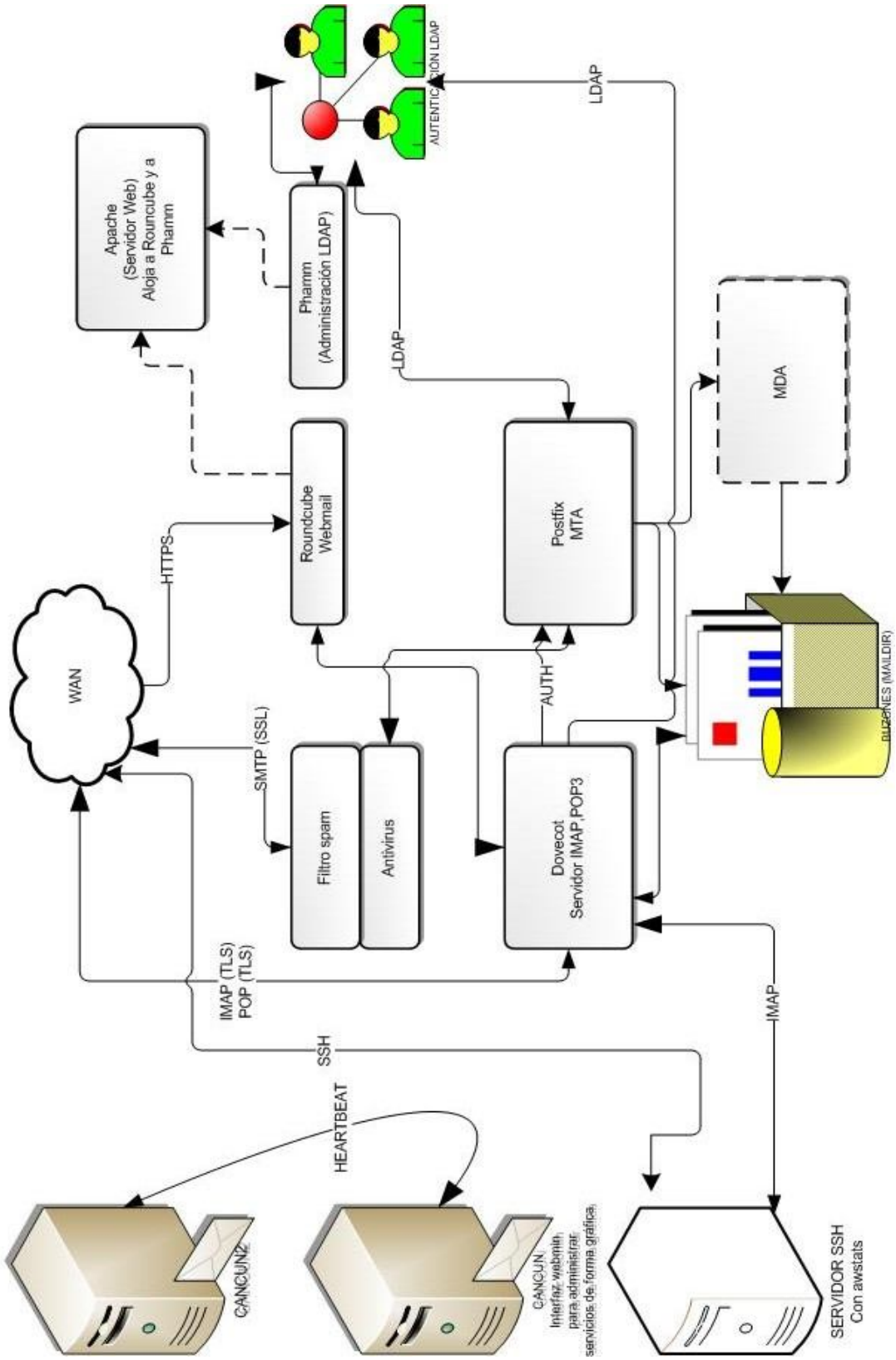


FIGURA 6.10 Implementación del Servicio de Correo en el Servidor piloto

Costos

En este apartado sólo se considera el costo de la implantación de la solución de correo debido a que la infraestructura ya existe (cableado, servidor, elementos de red, etc). La propuesta de correo electrónico incluye un sistema operativo (clon de red-hat), MTA, servidor IMAP y POP3, servidor Web con webmail, ldap para autenticación de usuarios y heartbeat como solución de alta disponibilidad. En la investigación de costos se tienen soluciones parecidas a la propuesta, es decir, soluciones que usen software libre para que no se incrementen costos debido a pagos de licencia.

De acuerdo a ciertas cotizaciones de empresas mexicanas dedicadas a la implantación y configuración de servidores utilizando software libre se recabaron lo siguientes datos.

Servidor de correo empresarial con antivirus $13,800 pesos para 20 usuarios
Usuarios adicionales $500 pesos más IVA por grupo de 5 usuarios

Por servicios unitarios se cobran 4500 pesos (aprox.) hasta 20 usuarios.

Con los datos anteriores podemos hacer una cotización tentativa del costo del sistema a implementar.

Servidor de Correo Empresarial con antivirus	$13 800
OpenLDAP	$ 4 500
Servidor Web y Webmail	$ 4 500
Servidor Imap/POP3	$ 4 500
Heartbeat	$ 4 500
Administración Gráfica	$ 4 500
Awstats	$ 4 500
Capacitación (5 personas)	$25 000 (5000 p/persona)
Subtotal	$65 800
IVA	$ 9 870
Total	$75 670 M. N.

El anterior cálculo se basa en un sistema con 20 usuarios, ahora, si lo que se quiere es hacer una estimación para 7000 mil usuarios se debe tomar en cuenta que por cada 5 usuarios se cobran $500 pesos, con esto se tiene:

Sistema base $13800 para 20 usuarios restan 6980 por cobrar, sale a 100 pesos por usuario.

6980*100=698 000 sólo por el sistema de correo con antivirus y antispam.

De los servicios restantes el que puede aumentar su precio es OpenLDAP ya que se tiene que agregar una entrada por cada usuario adicional, suponiendo que el aumento sea de $50 por usuario se tiene:
6980* 50=349 000

Ahora haciendo la cotización para 7000 usuarios.

Servidor de Correo Empresarial con antivirus	$ 13 800	
OpenLDAP	$ 4 500	
Usuarios adicionales	$ 698 000	
Servidor Web y Webmail	$ 4 500	
Servidor Imap/POP3	$ 4 500	
Heartbeat	$ 4 500	
Administración Gráfica	$ 4 500	
Awstats	$ 4 500	
Capacitación (5 personas)	$ 25 000	(5000 p/persona)
Subtotal	$ 763 800	
IVA	$ 114 570	
Total	$ 878 370 M. N.	

Existen soluciones de colaboración comerciales (correo electrónico, agenda, webmail, herramientas de comunicación interactiva) que utilizan software libre como la base de su solución.

Zimbra Collaboration Suite

Ofrece una solución de colaboración (Servidor de correo, agenda, chat, etc.). Se basa en software libre, utiliza postfix, OpenLdap, Clamav, amavis-new, jetty, etc. El costo de licencia se resume a continuación:
1 licencia por usuario cada año = $28.00 US

Es decir si se tienen 7000 usuarios el costo es de
$28 * 7000=$196000

Para instituciones educativas se tienen un descuento de 50% al año.
$196000/2= $98 000 US por año

Open-Xchange

Es una solución de colaboración que implementa estándares abiertos de internet.
Los costos son los siguientes:

Un año de subscripción al portal de mantenimiento (incluyendo las licencias para Open-Xchange server y los OXTender Microsoft Outlook, Palm OS y Active Directory) para 25 Usuarios (para instituciones académicas)-- $657 US

Usuario adicional $19.50 US

En 7000 usuarios tendríamos

$136 669.5 US

CONCLUSIONES Y COMENTARIOS FINALES

Ha llegado el punto de hacer una evaluación del trabajo, apoyándome en el objetivo inicial del proyecto puedo decir que el objetivo se cumplió ampliamente. El objetivo consistió en "Implantar un sistema de correo electrónico confiable, eficiente y escalable, tomando en cuenta mecanismos de seguridad, privacidad y alta disponibilidad que permita el intercambio de contenido diverso en los mensajes, utilizando Linux y una metodología de renovación".

De acuerdo a la implementación del servidor piloto puedo confirmar que todas las metas señaladas fueron cumplidas puesto que se implantó un servicio de correo electrónico confiable, eficiente y escalable, con mecanismos de seguridad, privacidad y alta disponibilidad, utilizando la reingeniería como herramienta de renovación e implementando el servicio sobre un sistema Linux.

Se cumplieron las demandas de los clientes referentes a espacio en disco, privacidad, seguridad y disponibilidad, así como de los administradores del sistema: facilidad en la administración, facilidad en el mantenimiento del sistema, portable entre distribuciones y más.

A continuación una breve explicación que muestra como se cubrieron los objetivos en las diferentes etapas del desarrollo del proyecto.

En la instalación del sistema operativo: se tomaron en cuenta mecanismos de seguridad como son la actualización de paquetes, deshabilitación de comandos y servicios innecesarios en el sistema, implementación de firewall de host y un detector de intrusos propio del sistema porque a pesar de que el departamento de seguridad de la Facultad analiza el flujo de datos en la red, al viajar los mensajes en un canal encriptado se dificulta el monitoreo de agentes maliciosos.

La instalación de los mecanismos de seguridad del sistema de correo contemplan la autenticación de usuarios en el envío de correos, también se tomó en cuenta el envío de información por canales encriptados (privacidad) para la mayoría de los protocolos manejados en el servidor.

La autenticación de los usuarios para el acceso a los buzones se realiza a través de un servicio ajeno al sistema, LDAP "*Lightweight Directory Acces Protocol*", para evitar que los usuarios interactúen directamente con el servidor, logrando reforzar la seguridad y aprovechando de una mejor forma los recursos haciendo al sistema más rápido.

Al aislar el servicio de acceso remoto se ahorra espacio en disco que puede ser dedicado a los buzones de correo aumentando el espacio de

almacenamiento de los mensajes de cada uno de los usuarios y los administradores se benefician porque tienen un servicio menos que vigilar.

El mecanismo de autenticación de LDAP tiene una interfaz de administración que facilita la operación del sistema. Se cuenta con un servicio de webmail para la recuperación de los correos a través de cualquier navegador lo que brinda movilidad.

Cada uno de los mensajes que procesa el sistema es analizado con un filtro de antivirus y antispam, aunque es recomendable que cada máquina que descargue correo de forma local tenga instalado software antivirus, cortafuegos y antispam.

Para facilitar el análisis de información de cada uno de los mensajes procesados por el sistema se tiene un software que realiza reportes detallados de los correos, a partir de estos análisis los administradores pueden tomar decisiones para mejorar el rendimiento general.
Finalmente, se implantó un mecanismo que busca la continuidad del sistema (Heartbeat, con dos nodos), de esta forma se tienen la alta disponibilidad del sistema de correo.

El diseño del sistema de correo está basado en software open-source, los paquetes elegidos cumplen con los requerimientos de estabilidad y seguridad propuestos para el sistema, además todos los paquetes han demostrado ser maduros en servidores de producción y pueden brindar servicios de manera distribuida si así lo requiere el sistema, es decir, se pueden implementar los servicios en equipos diferentes sin tener que rediseñar el sistema propuesto.

Cabe destacar que la instalación de la mayoría de los paquetes en el sistema de correo se realizó a través de código fuente y se compilaron para el sistema, con ello no existe problemas en la selección de alguna otra distribución de Linux o en la arquitectura del procesador.

El sistema está diseñado para ser rápido, estable, seguro y escalable. Sin embargo el éxito en la implantación final depende de la colaboración de todos los participantes en el proceso de desarrollo y el apoyo del líder del proyecto.

Los administradores deben capacitarse constantemente, se deben tener planes de administración así como manuales de operación del sistema. También se debe informar a los usuarios del servicio los puntos de mejora en el sistema para que aprovechen mejor el servicio e

informarles que a pesar de la seguridad implantada en el sistema, la protección más eficaz contra agentes maliciosos son la prevención y la cautela.

Todo sistema es susceptible de mejorarse, por lo que es recomendable hacer revisiones periódicamente. Una de las recomendaciones es implementar un mecanismo que permita la réplica de particiones en los nodos del clúster, esto se puede hacer con un software llamado “DRBD”.

Se debe buscar en lo posible separar los servicios para aumentar el rendimiento de cada uno de ellos, por ejemplo, separar el servicio web del correo aumentaría tanto el espacio en los buzones como en la velocidad del servicio. No se debe olvidar las actualizaciones de los paquetes de software y del sistema operativo.

BIBLIOGRAFÍA Y MESOGRAFÍA

CAPÍTULO 1

Sánchez, Sebastián. **Unix y Linux Guía Práctica**; Alfaomega. 1ª Edición. Madrid, España. 1999

*Whelan, Jonathan. **E-mail en el trabajo**. Prentice Hall.* 1ª Edición. *Madrid, España. 2000*

FUENTES DE INTERNET

Aprovechar el correo electrónico: Cómo funciona
http://www.learnthenet.com/spanish/html/20how.htm

Correo Electrónico
http://www.uv.es/ciuv/cas/correo/email.html

¿Qué es reingeniería?
http://www.kyna.com.mx/Que_es_Reingenieria.htm

Reingeniería
http://www.monografias.com/trabajos/reingenieria/reingenieria.shtml

Reingeniería
http://intraremington.remington.edu.co/admon/und5rein.htm

Todo sobre el correo electrónico: el E-mail
http://usuarios.lycos.es/nachos/elemail.htm

Capítulo 2

CETTICO**. Teleinformática**, Enciclopedia de Informática y Computación, Cultural S.A. 1ª Edición. España, 1997

García, Jesús, Raya, Luis y Raya, Víctor. **Alta Velocidad y calidad de servicio en redes IP**. Alfaomega. 1ª Edición, Madrid, España. 2002

Hunt, Craig. **TCP/IP Network Administration**, O'Reilly. 3ª Edición, E.U. 2002

López, Ángel y Novo, Alejandro. **Protocolos de Internet, Diseño e implementación en sistemas UNIX**. Alfaomega. 1ª Edición, Madrid, España. 2000

Microsoft, **Fundamentos de Redes Plus, curso oficial de certificación MCSE**. McGraw-Hill. 1ª Edición, E.U. 2000

FUENTES DE INTERNET

RFC HTTP 2616
http://www.ietf.org/rfc/rfc2616.txt

RFC IMAP 3521
http://www.faqs.org/rfcs/rfc3501.html

RFC POP 1939
http://www.ietf.org/rfc/rfc1939.txt

RFC SMTP 821
http://www.faqs.org/rfcs/rfc821.html

Capítulo 3

Costales, Bryan & Allman, Eric. **Sendmail**, O'Reilly. 2ª Edición, E.U. 1997

Dent, Kyle. **Postfix: The Definitive Guide**, O'Reilly. 1ª Edición, E.U. 2003

Hazel, Philip. **Exim: The Mail Transfer Agent**, O'Reilly. 1ª Edición, E.U. 2001

Sill, Dave. **The qmail Handbook**, Apress. 1ª Edición. E.U. 2002

FUENTES DE INTERNET

Allman, E. Assman, Neil. **Sendmail: Installation and Operation Guide**, Sendmail Inc.

http://www.sendmail.org/doc/sendmail-current/doc/op/op.pdf

Exim
www.exim.org

Maildrop
http://www.courier-mta.org/maildrop/

Outlook
http://office.microsoft.com/es-es/outlook/default.aspx

Pine
http://www.washington.edu/pine/

Postfix
www.postfix.org

Procmail
www.procmail.org

qmail
www.qmail.org

Sendmail
www.sendmail.org

Capítulo 4

Bauer, Michael. **Linux Server Security**, O'Reilly. 2ª Edición, E.U. 2005
Linux Máxima Seguridad, Prentice Hall. 2002. 1ª Edición, España

FUENTES DE INTERNET

Guidelines on Electronic Mail Security
http://csrc.nist.gov/publications/nistpubs/800-45-version2/SP800-45v2.pdf

Hoaxes
http://www.rompecadenas.com.ar/hoaxlist.htm

La importancia del análisis de Malware
http://www.malware.unam.mx/documentos.dsc

Pretty Good Privacy
http://www.pgpi.org

Capítulo 5

Carter, Gerald. **LDAP System Administration**, O'Reilly. 1ª Edición. E.U. 2003

Voglmaier, E. **The ABCs of LDAP: How to Install, Run, and Administer LDAP Services**, CRC Press, E.U. 2004

FUENTES DE INTERNET

High Availability Cluster for Linux
http://www.linuxjournal.com/article/3247

Linux.com :: Keep your Web site online with a *High Availability*
http://www.linux.com/feature/57362

Capítulo 6

Xue, Jack. **High-Availability E-mail System with Active Directory**, Linux Journal. E.U. November 2007

Bartholomew, Daniel. **Getting Started with Heartbeat,** Linux Journal. E.U. November 2007

FUENTES DE INTERNET

dovecot
www.dovecot.org

HOWTO: *Postfix*, Dovecot, *Jamm*, OpenLDAP, SSL, and SASL
http://wanderingbarque.com/howtos/mailserver/mailserver.html

Linux-Ha
http://www.linux-ha.org/

Phamm
www.phamm.org

Servidor de Correo con dominios virtuales sobre un directorio LDAP
http://www.tuxjm.net/docs/mailserver-howto/