Universidad Nacional Autónoma de Nicaragua.
UNAN – León.
Facultad de Ciencias.
Departamento de Computación.

Monografía para optar al titulo de:

Ingeniero en Sistemas de Información.

Configuración y Administración de Open-Xchange Server bajo la plataforma Linux.

Presentado por:

Br. Juan Carlos Bordas Montoya.
Br. Arllen Javier Díaz Cáceres.
Br. Nubia Consuelo Espinoza García.

Tutor:
MSc. Aldo Rene Martínez.

León, martes, 22 de marzo de 2011.

ÍNDICE.

I. DEDICATORIA.

Juan Carlos Bordas Montoya.

Este trabajo monográfico lo dedico primeramente a Dios ya que me ha permitido finalizarlo.

A mis padres quienes me han brindado su confianza, apoyo y ayuda incondicional en todas mis decisiones.

Arllen Javier Díaz Cáceres.

Esta monografía la dedico en primer lugar a Dios que es el que me ha dado el conocimiento, las fuerzas y sobre todo la vida para culminar este trabajo.

A mi madre Maria de la Concepción Cáceres, quien me ha apoyado siempre en mis proyectos, y en la vida diaria y a mi padre y hermanos por su apoyo incondicional.

Nubia Consuelo Espinoza García.

Dedico esta monografía principalmente a Dios por haberme dado la vida y haber permitido llegar al final de este trabajo.

A mis padres Consuelo García y Rider Espinoza por haberme brindado su confianza y su apoyo incondicional durante esta etapa de mi vida.

A mis hermanos.

II. AGRADECIMIENTO.

Estamos profundamente agradecidos con Dios (Jehová), que nos da fuerza y conocimiento para llegar hasta el final de nuestras metas.

Estamos agradecidos por el gran apoyo paciencia y dedicación que nos ha dado el MSc. Aldo Rene Martínez.

Al licenciado Edisón Cuevas por su apoyo, colaboración y disposición a pesar de la distancia.

A nuestros padres por el apoyo incondicional que nos mostraron durante todo este periodo monográfico.

A todas las personas que hicieron posible la culminación de este proyecto.

III. INTRODUCCIÓN.

Este sistema colaborativo para trabajo en grupo (OX) se va a implementar en el Departamento de Computación, dado que en la actualidad no se cuenta con una herramienta de este tipo tan necesarias en las empresas actuales. Una vez implementado se promoverá entre los docentes del Departamento el trabajar en grupo, explicando las ventajas que este implique.

Open-Xchange es una aplicación para el trabajo en equipo (del inglés groupware) o sistema colaborativo (collaborative software) que proporciona a sus usuarios un avanzado sistema de comunicaciones y funciones para la colaboración.

Groupware se refiere a los programas informáticos que integran el trabajo de un proyecto con muchos usuarios concurrentes que se encuentran en diversas estaciones de trabajo, típicamente conectados a través de una red Internet o de una intranet.

Las características de la mensajería incluyen correo electrónico, filtro anti-spam y detector de virus. Los aspectos más básicos de colaboración incluyen: el calendario, los contactos, la gestión de tarea y carpetas privadas, públicas y compartidas. Los aspectos más avanzados son: la completa integración mediante enlaces y permisos con la compartición de documentos, el seguimiento de proyectos, el repositorio de marcadores o favoritos, el tablón de anuncios, los foros de debate y el archivo de conocimiento. Todo ello a través de una vista integrada a modo portal o página de inicio.

Open-Xchange Server es software libre y de código abierto, y se distribuye bajo la licencia GPL (General Public License), creado sobre estándares abiertos e internacionales, lo que significa el uso de API´s estándar, protocolos estándar y formatos de datos estándar. Algunos de los estándares que utiliza son: POP3, IMAP4, HTML, XML, JavaScript, LDAP y SQL.

IV. ANTECEDENTES.

En la Facultad de Ciencias de la UNAN-León es la encargada de ofrecer las carreras de informática, por lo tanto tiene el interés de ir de la mano con los avances tecnológicos.

En estos últimos años las instituciones están emigrando y optando por tener este tipo de sistema colaborativo Open-Xchange debido a los bajos costos (por no decir nulos) que implican. Además la universidad se encuentra en una transformación de sus servicios que antes eran por licencia pagada para que de ahora en adelante sean bajo licencia GPL.

En años anteriores el Departamento de Computación en colaboración con los alumnos han visto la necesidad, la utilidad y el mejor rendimiento que se obtiene utilizando el código libre, por lo tanto se han elaborado un sinnúmero de proyectos basados en estas licencias, por lo que nosotros somos continuadores de esta tendencia.

En el Departamento hasta el momento no se ha implementado un sistema colaborativo basado en licencia GPL, pero en la UNAN-Managua si se ha utilizado, por lo tanto somos pioneros en instalar y administrar el sistema Open-Xchange en esta Universidad.

V. JUSTIFICACIÓN.

Este proyecto se realiza con el objetivo de dar a conocer las grandes posibilidades de conectividad que Open-Xchange Server ofrece, como acceso universal, permite el trabajo en equipo desde cualquier lugar, sobre cualquier tipo de red de comunicaciones y utilizando cualquier dispositivo de comunicaciones disponibles.

Es una solución corporativa dado que aumenta la productividad (a través del trabajo en equipo), reduce costos y máxima la flexibilidad técnica.

Dado que las empresas necesitan un sistema colaborativo a muy bajo costo, se realizo este trabajo como una alternativa a Microsoft Exchange, el cual es de muy alto costo.

Open-Xchange Server es útil porque es una plataforma Web rentable construida con estándares abiertos de fuente optimizada para compañías pequeñas y grandes, permitiendo a sus empleados de todo el mundo comunicar e intercambiar rápida y eficientemente la información usando apenas un navegador; los empleados pueden tener acceso a todos sus e-mail así como a su depósito de documentos, tareas, contactos, calendario, favoritos en cuestión de segundos, sin importar su localización física.

Nosotros elegimos OX como sistema colaborativo debido a que está probado en diferentes escenarios y por miles de usuarios en todo el mundo, es la inspiración de centeneras de programadores que hacen evolucionar más rápidamente todas sus funcionalidades, asegurando todas las expectativas de desarrollo.

Además de las justificaciones antes mencionadas la Facultad de Ciencias no cuenta con un sistema de trabajo en grupo, el cual urge para el aumento de las capacidades laborales de los docentes.

VI. OBJETIVOS.

Objetivo General.

- Habilitar un sistema colaborativo en la Facultad de Ciencias que proporcione una plataforma para el trabajo en grupo de docentes y administrativos.

Objetivo Específicos.

- Instalar y configurar un servidor que corra los servicios DNS, Web, Correo electrónico, LDAP, Open-Xchange bajo la plataforma Suse 10.2.
- Proporcionar una documentación fidedigna acerca de la configuración de Open-Xchange, DNS, Apache, Tomcat, LDAP, Postfix, Cyrus.
- Elaborar una documentación descriptiva de los aspectos fundamentales en que se basa Open-Xchange.
- Utilizar sistemas basados en licencias GPLs.
- Facilitar herramientas administrativas Web para los servidores DNS, Web, Correo electrónico, LDAP, Open-Xchange, PostgreSQL.

VII. MARCO TEÓRICO.

Capa de aplicación.

Esta capa describe como hacen su trabajo los programas de aplicación (navegadores, clientes de correo, terminales remotos, transferencia de ficheros etc.).

Ofrece a las aplicaciones (de usuario o no) la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (POP y SMTP), gestores de bases de datos y servidor de ficheros (FTP). Hay tantos protocolos como aplicaciones distintas y puesto que continuamente se desarrollan nuevas aplicaciones el número de protocolos crece sin parar.

Cabe aclarar que el usuario normalmente no interactúa directamente con el nivel de aplicación. Suele interactuar con programas que a su vez interactúan con el nivel de aplicación pero ocultando la complejidad subyacente.

Es el nivel más cercano al usuario y a diferencia de los demás niveles, por ser el más alto o el último, no proporciona un servicio a ningún otro nivel. Cuando se habla de aplicaciones lo primero que viene a la mente son las aplicaciones que procesamos, es decir, nuestra base de datos, una hoja de cálculo, un archivo de texto, etc, lo cual tiene sentido ya que son las aplicaciones que finalmente deseamos transmitir.

Sin embargo, en el contexto del Modelo de Referencia de Interconexión de Sistemas Abiertos (OSI), al hablar del Nivel de Aplicación no nos estamos refiriendo a las aplicaciones que acabamos de citar. En OSI el nivel de aplicación se refiere a las aplicaciones de red que vamos a utilizar para transportar las aplicaciones del usuario.

DNS (Domain Name System).

Protocolo DNS.

Llamamos DNS al protocolo de comunicación entre un cliente (resolver) y el servidor DNS.

El protocolo DNS está compuesto por dos programas uno llamado servidor de nombres de dominios y otro llamado resolvers. Los servidores de nombres de dominios contienen la base de datos de un segmento y dicha base de datos es accesada por los clientes a través de un programa conocido como resolvers. Los resolvers son rutinas utilizadas para tener acceso a la base de datos ubicada en los servidores de nombres de dominios con el fin de resolver la búsqueda de una dirección IP asociada a un nombre.

Introducción a DNS.

Es una base de datos distribuida y jerárquica que almacena información relativa a los nombres de dominio en Internet o gestiona nombres de equipos y servicios en redes locales. El uso más común de una base de datos DNS es la de asignación de nombres de dominio o de servidores de correo a direcciones IP.

Dicha asignación se utilizará para la localización de dichos equipos/servicios de una manera sencilla y sin tener que recordar cada vez la dirección real. La información dada se puede consultar a la inversa (una dirección IP se traduce en un nombre almacenado en la base de datos).

Los componentes principales de un sistema DNS son los siguientes:

- Clientes DNS: Encargados de realizar las consultas pertinentes a las bases de datos de los servidores DNS.

- Servidores DNS: Contestan a las peticiones realizadas por los clientes. Dicha contestación se hará consultando la base de datos propia o haciendo una consulta recursiva a otro servidor DNS.

- Zonas de autoridad: Son los espacios de nombres de dominio donde se almacenan los datos. Generalmente, una zona de autoridad comprende, al menos, un nombre de dominio y todos sus subdominios, pudiendo estos últimos estar en sus zonas de autoridad propias.

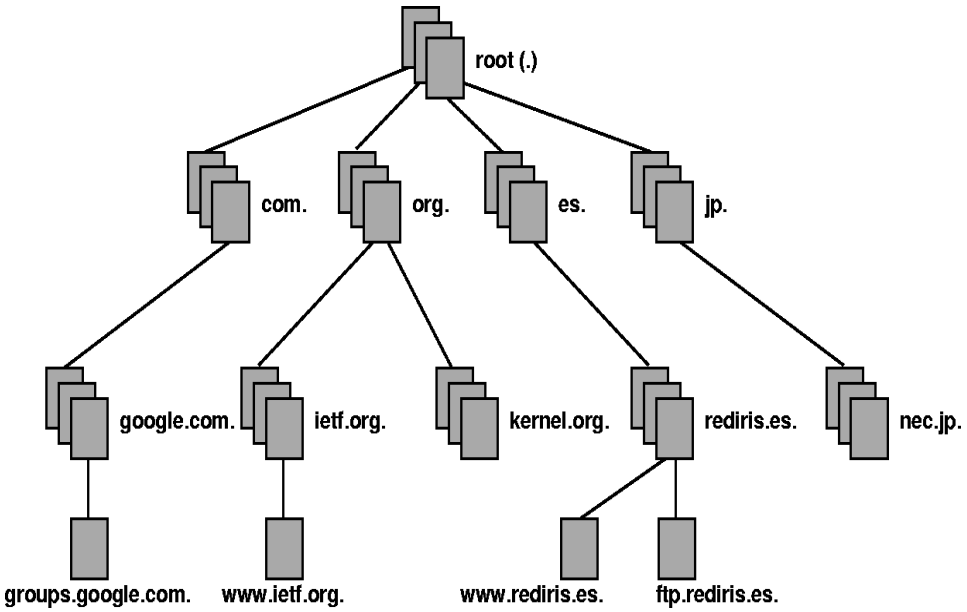

Figura 1: Parte del espacio de nombres de dominio de Internet.

Servidores DNS.

Un servidor DNS es capaz de recibir y resolver peticiones relacionadas con el sistema de nombres. Un servidor DNS sirve, por tanto, para traducir un nombre de dominio en una dirección IP, asignar nombres a todas las máquinas de una red y trabajar con nombres de dominio en lugar de IPs. El uso de este servidor es transparente para los usuarios cuando éste está bien configurado.

Cada LAN (Red de área local) debería contar con un servidor DNS. Estos servidores trabajan de forma jerárquica para intercambiar información y obtener las direcciones IP de otras LANs.

Tipos de Consultas.

Consultas Iterativas: El servidor DNS responde a la consulta del cliente desde los datos guardados en su base de datos o en las bases de los sistemas locales. Si dicha información no se encuentra disponible, la petición se reenviará hacia otros servidores, hasta encontrar la Zona de Autoridad válida que resuelva la petición. En principio, la carga de la consulta recae sobre el cliente.

Consultas Recursivas: El servidor DNS proporciona, si existe, la respuesta a la solicitud. Para ello, hará tantas consultas iterativas como sea necesario hasta dar con el dato solicitado. Las peticiones son transparentes a la máquina del cliente.

Zonas de autoridad.

Un servidor DNS Primario carga su información desde una Zona de Autoridad. Dicha zona abarca un nombre de dominio y, si los nombres de los subdominios no están delegados, también los incluirá. Toda la información se almacenará localmente en un fichero que contendrá alguno de estos tipos de registros:

- **A (Address):** Registro de dirección que resuelve un nombre de un anfitrión hacia una dirección IPv4 de 32 bits.

- **CNAME (Canonical Name):** Registro de nombre canónico que hace que un nombre sea alias de otro. Los dominios con alias obtienen los sub-dominios y registros DNS del dominio original.

- **MX (Mail Exchanger):** Registro de servidor de correo que sirve para definir una lista de servidores de correo para un dominio, así como la prioridad entre éstos.

- **PTR (Pointer):** Registro de apuntador que resuelve direcciones IPv4 hacia el nombre anfitriones. Es decir, hace lo contrario al registro A. Se utiliza en zonas de Resolución Inversa.

- **NS (Name Server):** Registro de servidor de nombres que sirve para definir una lista de servidores de nombres con autoridad para un dominio.

- **SOA (Start of Authority):** Registro de inicio de autoridad que especifica el Servidor DNS Maestro (o Primario) que proporcionará la información con autoridad acerca de un dominio de Internet, dirección de correo electrónico del administrador, número de serie del dominio y parámetros de tiempo para la zona.

- **SRV (Service):** Registro de servicios que especifica información acerca de servicios disponibles a través del dominio. Protocolos como SIP (*Session Initiation Protocol*) y XMPP (*Extensible Messaging and Presence Protocol*) suelen requerir registros SRV en la zona para proporcionar información a los clientes.

- **TXT (Text):** Registro de texto que permite al administrador insertar texto arbitrariamente en un registro DNS. Este tipo de registro es muy utilizado por los servidores de listas negras DNSBL (*DNS-based Blackhole List*) para la filtración de Spam. Otro ejemplo de uso son las VPN, donde suele requerirse un registro TXT para definir una llave que será utilizada por los clientes.

Las zonas a resolver serán las siguientes:

- **Zonas de Reenvío:** Devuelven direcciones IP para búsquedas sobre nombres FQDN (Fully Qualified Domain Name). Es importante apuntar aquí que, en el caso de tratarse de dominios públicos, hay una responsabilidad por parte de la autoridad misma del dominio (el Registrar del WHOIS) para crear dicha zona de reenvío.

- **Zonas de Resolución Inversa:** Devuelven nombres FQDN para búsquedas sobre direcciones IP. Como en el caso anterior, la responsabilidad de crear la Zona de Autoridad recae sobre la autoridad misma del segmento (si hacemos un WHOIS sobre una dirección IP, obtendremos a la autoridad de todo el segmento de direcciones.

Inicio del servidor DNS (Bind).

En un sistema SUSE Linux, el servidor de nombres BIND (Berkeley Internet Name Domain) viene configurado previamente, de manera que puede iniciarse justo después de la instalación sin ningún problema. Si ya hay una conexión de Internet y ha introducido 127.0.0.1 como la dirección del servidor de nombres para localhost en /etc/resolv.conf, por lo general ya tendrá una resolución de nombres en funcionamiento sin tener que saber el DNS del proveedor.

BIND lleva a cabo una resolución de nombres mediante el servidor de nombres raíz, un proceso mucho más lento. Por norma general, debería introducirse el DNS del proveedor con su dirección IP en el archivo de configuración /etc/named.conf en la línea forwarders para asegurar una resolución de nombres efectiva y segura.

Si todo esto funciona hasta ahora, el servidor de nombres se ejecutará como un servidor de nombres sólo para almacenamiento en caché. Únicamente cuando configure sus propias zonas, se convertirá en un DNS adecuado. En la documentación de /usr/share/doc/packages/bind/sample-config. Se muestra un ejemplo sencillo de todo lo explicado hasta ahora.

Para iniciar el servidor de nombres, introduzca el comando rcnamed start o /etc/init.d/named start como usuario Root. Si a la derecha aparece “done” (finalizado) en verde, querrá decir que "named", que es como se denomina al proceso del servidor de nombres, se ha iniciado correctamente. Compruebe el servidor de nombres inmediatamente en el sistema con los programas host, dig o nslookup.

Si aparece un mensaje de error, utilice rcnamed status para ver si el servidor se está ejecutando realmente. Si el servidor de nombres no se inicia o se comporta de manera inesperada, normalmente podrá encontrar la causa en el archivo de registro /var/log/messages.

Nota: Adaptación automática del DNS (con DHCP en el caso más general):
Dependiendo del tipo de conexión a Internet o a la red, la información del servidor de nombres puede adaptarse automáticamente a las condiciones actuales.

Para hacerlo, defina la variable MODIFY_NAMED_CONF_DYNAMICALLY del archivo /etc/sysconfig/network/config en yes.

Servidor Web.

Un servidor Web es un programa que implementa el protocolo HTTP (Protocolo de Transferencia Hipertexto). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (Hypertext Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

El servidor Web se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP que solemos conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. Sobre el servicio Web clásico podemos disponer de aplicaciones Web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

Aplicaciones en el lado del cliente: el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta.

Aplicaciones en el lado del servidor: el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

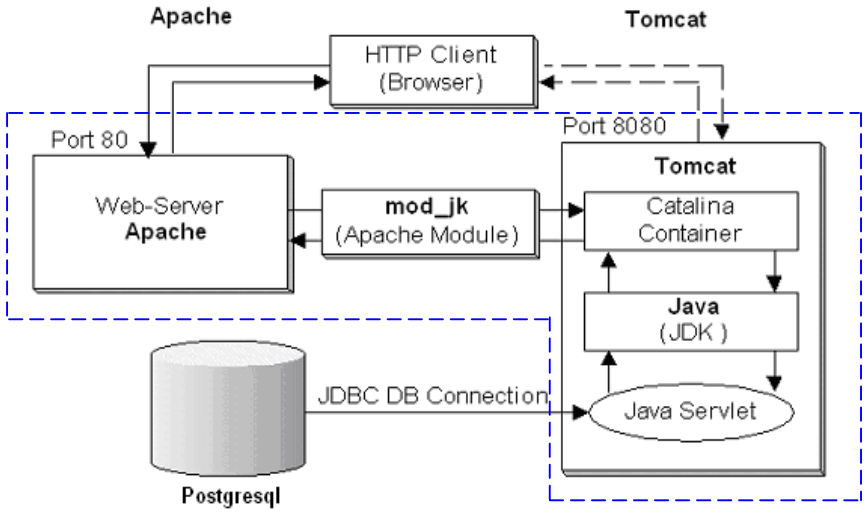

Figura 2: Comunicación entre Servidores Web con acceso a base de datos.

Protocolo HTTP (Hyper Text Transfer Protocol).

Es el protocolo usado en cada transacción de la Web (WWW). HTTP fue desarrollado por el consorcio W3C y la IETF, colaboración que culminó en 1999 con la publicación de una serie de RFCs, siendo el más importante de ellos el RFC 2616, que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura Web (clientes, servidores, proxies) para comunicarse.

Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador o un spider) se lo conoce como "user agent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un URL. Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones Web necesita frecuentemente mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones Web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.

Etapas de una transacción HTTP.

Cada vez que un cliente realiza una petición con un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo Location del cliente Web.

- El cliente Web decodifica la URL, separando sus diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o ip del servidor, el posible puerto opcional (80) y el objeto requerido del servidor.

- Se habré una conexión TCP / IP con el servidor, llamando al puerto TCP correspondiente.

- El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno seguido de la propia información.

- Se cierra la conexión TCP.

Apache.

Apache es un servidor Web gratuito, potente y que nos ofrece un servicio estable y sencillo de mantener y configurar, se encarga de resolver las peticiones de páginas de Internet de los clientes utilizando el protocolo de Internet HTTP. Es indiscutiblemente uno de los mayores logros del Software Libre, esta hecho para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual.

Características de Apache.

- ✓ Es multiplataforma, aunque idealmente está preparado para funcionar bajo linux.
- ✓ Muy sencillo de configurar.
- ✓ Es Open Source (código abierto).
- ✓ Muy útil para proveedores de Servicios de Internet que requieran miles de sitios pequeños con páginas estáticas.
- ✓ Amplias librerías de PHP y Perl a disposición de los programadores.
- ✓ Posee diversos módulos que permiten incorporarle nuevas funcionalidades, estos son muy simples de cargar.
- ✓ Es capaz de utilizar lenguajes como PHP, TCL, Python y Perl.
- ✓ Módulos de autenticación: mod_access, mod_auth y mod_digest.
- ✓ Soporte para SSL y TLS.
- ✓ Permite la configuración de mensajes de errores personalizados y negociación de contenido.
- ✓ Permite autenticación de base de datos basada en SGBD.

Uso de Apache.

Apache es principalmente usado para servir páginas Web estáticas y dinámicas en la WWW, es el servidor Web del popular sistema XAMP, junto con MySQL y los lenguajes de programación PHP/Perl/Python. La "X" puede ser la inicial de cualquier sistema operativo, si es Windows: WAMP, si es el Linux: LAMP, etc.

Requisitos.

- ✧ La red del equipo debe de estar configurada correctamente.
- ✧ La hora exacta del sistema del equipo se debe de sincronizar con un servidor horario, lo que es preciso debido a que ciertas partes del protocolo HTTP dependen de que la hora sea correcta.
- ✧ Que estén instaladas las últimas actualizaciones de seguridad.
- ✧ Que el puerto de servidor Web por defecto (puerto 80) este abierto en el corta fuegos. Para ello, configure SUSEFirewall2 para que se permita el servicio Servidor HTTP en la zona externa.

Ventajas.

- ☆ Modular.
- ☆ Open Source.
- ☆ Multi-plataforma.
- ☆ Extensible.
- ☆ Popular (fácil conseguir ayuda/soporte).
- ☆ Gratuito.

Módulos.

La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor Web. Algunos de estos módulos son:

- ❋ mod_ssl - Comunicaciones Seguras vía TLS.
- ❋ mod_rewrite - reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a los navegantes o a los motores de búsqueda en cuanto a como fueron desarrolladas estas páginas).
- ❋ mod_dav - Soporte del protocolo WebDAV (RFC 2518).
- ❋ mod_deflate - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- ❋ mod_auth_ldap - Permite autentificar usuarios contra un servidor LDAP.
- ❋ mod_proxy_ajp - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets y JSP).

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- mod_perl - Páginas dinámicas en Perl.
- mod_php - Páginas dinámicas en PHP.
- mod_python - Páginas dinámicas en Python.
- mod_rexx - Páginas dinámicas en REXX y Object REXX.
- mod_ruby - Páginas dinámicas en Ruby.
- mod_aspdotnet - Páginas dinámicas en .NET_de_Microsoft (Módulo retirado).
- mod_mono - Páginas dinámicas en Mono.
- mod_security - Filtrado a nivel de aplicación, para seguridad.

Tomcat.

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Un contenedor de Servlets es un shell de ejecución que maneja e invoca servlets por cuenta del usuario. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Tomcat es un servidor Web con soporte de servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor Web Apache. Tomcat puede funcionar como

servidor Web por sí mismo. Tomcat es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Podemos dividir los contenedores de Servlets en:

- Contenedores de Servlets Stand-alone (Independientes). Estos son una parte integral del servidor Web. Este es el caso cuando se usa un servidor Web basado en Java, por ejemplo, el contenedor de servlets es parte de JavaWebServer. Este el modo por defecto usado por Tomcat.

- Contenedores de Servlets dentro de proceso. Es una combinación de un Plugin para el servidor Web y una implementación de contenedor Java. El Plugin del servidor Web abre una JVM (Máquina Virtual Java) dentro del espacio de direcciones del servidor Web y permite que el contenedor Java se ejecute en él. Si una cierta petición debería ejecutar un servlet, el Plugin toma el control sobre la petición y lo pasa al contenedor Java (usando JNI). Un contenedor de este tipo es adecuado para servidores multi-thread de un sólo proceso y proporciona un buen rendimiento pero está limitado en escalabilidad.

- Contenedores de Servlets fuera de proceso. Es una combinación de un Plugin para el servidor Web y una implementación de contenedor Java que se ejecuta en una JVM fuera del servidor Web. El Plugin del servidor Web y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP). Si una cierta petición debería ejecutar un servlet, el Plugin toma el control sobre la petición y lo pasa al contenedor Java.

Tomcat puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como Plugin para un servidor Web existente (actualmente se soportan los servidores Apache, IIS y Netscape).

La configuración de Tomcat se basa en dos ficheros:

- **Server.xml:** El fichero de configuración global de Tomcat. Sirve para dos objetivos:

    - Proporcionar configuración inicial para los componentes de Tomcat.
    - Especifica la estructura de Tomcat, es decir permite que Tomcat arranque y se construya a sí mismo ejemplarizando los componentes especificados en server.xml.

Los elementos más importantes de server.xml:

- **Server:** Define un servidor Tomcat. Generalmente no deberíamos tocarlo demasiado. Un elemento Server puede contener elementos Logger y ContextManager.

- **Logger:** Define un objeto logger. Cada objeto de este tipo tiene un nombre que lo identifica, así como un path para el fichero log que contiene la salida y un verbosityLevel (que especifica el nivel de log).

- **ContextManager:** Especifica la configuración y la estructura para un conjunto de ContextInterceptors, RequestInterceptors, Contexts y sus Connectors. El ContextManager tiene unos pocos atributos que le proporcionamos con:

- Nivel de depuración usado para marcar los mensajes de depuración.

- La localización base para webapps/, conf/, logs/ y todos los contextos definidos. Se usa para arrancar Tomcat desde un directorio distinto a TOMCAT_HOME.

- El nombre del directorio de trabajo.

- Se incluye una bandera para controlar el seguimiento de pila y otra información de depurado en las respuestas por defecto.

- **Connector:** El Connector representa una conexión al usuario, a través de un servidor Web o directamente al navegador del usuario (en una configuración independiente). El objeto connector es el responsable del control de los threads en Tomcat y de leer/escribir las peticiones/respuestas desde los sockets conectados a los distintos clientes. La configuración de los conectores incluye información como:

    - La clase handler.

    - El puerto TCP/IP donde escucha el controlador.

    - El backlog TCP/IP para el server socket del controlador.

- **Context:** Cada Context representa un path en el árbol de tomcat donde situamos nuestra aplicación Web. Un Context Tomcat tiene la siguiente configuración:

    - El path donde se localiza el contexto. Este puede ser un path completo o relativo al home del ContextManager.

    - Nivel de depuración usado para los mensajes de depuración.

    - Una bandera reloadable. Cuando se desarrolla un servlet es muy conveniente tener que recargar el cambio en Tomcat, esto nos permite corregir errores y hacer que Tomcat pruebe el nuevo código sin tener que parar y arrancar. Para volver a recargar el servlet seleccionamos la bandera reloadable a true. Sin embargo, detectar los cambios consume tiempo; además, como el nuevo servlet se está cargando en un nuevo objeto class-loader hay algunos casos en los que esto lanza errores de forzado (cast). Para evitar estos problemas, podemos seleccionar la bandera reloadable a false, esto desactivará esta característica.

- Web.xml: Configura los distintos contextos en Tomcat.

Correo Electrónico.

El e-mail o correo electrónico es un servicio de red que funciona sobre Internet, que permite enviar y recibir mensajes (y todo tipo de documentos) de manera instantánea.

Básicamente, el e-mail es un servicio que copia un fichero de una máquina a otra y lo añade al buzón de correo mailbox/maildir del destinatario. Pero no es un asunto sencillo: el envío de un mensaje de correo es un proceso largo y complejo, en el que intervienen varios programas trabajando en cadena para conseguir que el e-mail llegue a su destino.

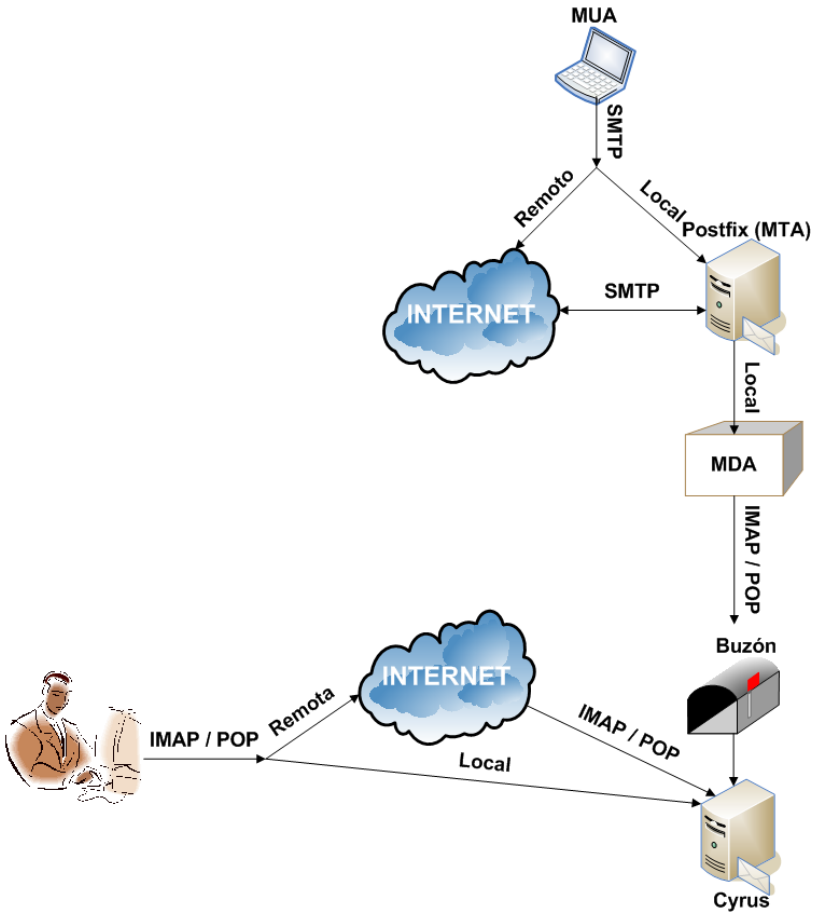

Figura 3: Sistema de correo electrónico.

Del gráfico anterior podemos comprender que un servidor de mail está compuesto por varios componentes, resumiendo: MTA, MDA y Servidor POP3/IMAP.

MUA (Mail User Agent).

Un MUA es un programa que permite a un usuario, como mínimo, leer y escribir mensajes de correo electrónico. A un MUA se le denomina a menudo cliente de correo. Lógicamente, hay muchos programas MUA que ofrecen al usuario muchas más

funciones, entre las que se incluyen la recuperación de mensajes mediante los protocolos POP e IMAP, la configuración de buzones de correo para almacenar los mensajes o ayuda para presentar los mensajes nuevos a un programa MTA (Mail Transfer Agent, Agente de transferencia de correo) que los enviará al destino final.

Los programas MUA pueden ser gráficos, como Mozilla Mail, o pueden tener una interfaz basada en texto sencilla, como Mutt o Pine.

MTA (Mail Transfer Agent).

Un programa MTA transfiere los mensajes de correo electrónico entre máquinas que usan el protocolo SMTP. Un mensaje puede pasar por varios MTA hasta llegar al destino final. La mayoría de los usuarios desconocen la existencia de estos agentes, incluso si cada mensaje se envía a través de como mínimo un MTA.

Aunque el proceso de envío de mensajes entre las máquinas puede parecer bastante directo, todo el proceso de decidir si un agente MTA concreto puede o debe aceptar un mensaje para entregarlo a un host remoto es bastante complicado. Además, debido a los problemas de correo basura, el uso de un MTA concreto normalmente está limitado por la propia configuración del MTA o el acceso a la red del sistema que lo ejecuta.

Muchos de los agentes MUA de mayores dimensiones y complejidad también sirven para enviar correo. Sin embargo, no se debe confundir esta acción con las funciones propias y verdaderas de estos agentes. Para que los usuarios que no ejecutan un agente MTA propio puedan transmitir los mensajes salientes a una máquina remota para su envío, deben utilizar una capacidad en el MUA capaz de transferir el mensaje a un MTA para el que tengan autorización de uso. Sin embargo, el agente MUA no entrega directamente el mensaje al servidor de correo del destinatario final; esta función está reservada al agente MTA.

Red Hat Linux utiliza Sendmail como agente MTA por defecto, aunque se pueden utilizar otros muchos en su lugar. Es importante desactivar el uso del agente MTA que se esté ejecutando actualmente antes de ejecutar otro, ya que ambos tratarán de utilizar el puerto 25, el puerto SMTP por defecto.

MDA (Mail Delivery Agent).

Los agentes MTA utilizan programas MDA para entregar el correo electrónico al buzón de un usuario concreto. En muchos casos, el agente MDA es realmente un LDA (Local Delivery Agent, Agente de entrega local), como bin/mail o Procmail. Sin embargo, Sendmail también puede desempeñar la función de un agente MDA, como cuando acepta un mensaje de un usuario local y lo adjunta a su fichero de spool de correo electrónico.

Cualquier programa que gestione realmente un mensaje para entregarlo al punto donde lo leerá un agente MUA se puede considerar un agente MDA. Tenga en cuenta que los agentes MDA no transportan mensajes entre sistemas ni actúan como interfaz para el usuario final.

Muchos usuarios no utilizan directamente agentes MDA, porque sólo se necesitan agentes MTA y MUA para enviar y recibir correo. Sin embargo, algunos agentes MDA se

pueden utilizar para ordenar los mensajes antes de que los lea el usuario, lo cual es de gran ayuda si recibe una gran cantidad de correo.

Entrega Final.

El correo electrónico se entrega al hacer que el emisor establezca una conexión TCP con el receptor y después que envié el correo electrónico a través de ella. Este modelo funciono bien por década cuando todo los host ARPANET (y mas tarde Internet) se pusieron, de hecho, en línea todo el tiempo para aceptar conexiones TCP.

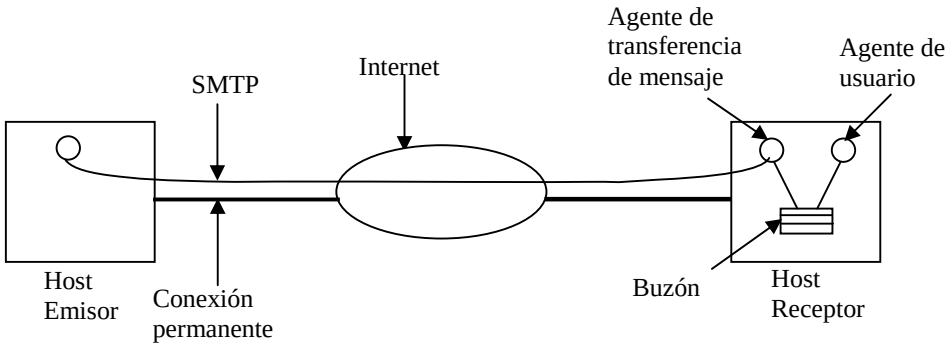

Figura 4: Envío y recepción de correo cuando el receptor tiene una conexión permanente a Internet y el agente de usuario se ejecuta en la misma maquina que el agente de transferencia.

Sin embargo, con el advenimiento de personas que acceden a Internet llamando a su ISP por medio de un MODEM ese modelo dejo de usarse. El problema es el siguiente:

¿Qué sucede cuando “X” persona desea enviar correo electrónico a “Y” persona y esta ultima no esta en línea en este momento?
Esa “X” persona no puede establecer una conexión TCP con la “Y” persona y por lo tanto no puede ejecutar el protocolo SMTP.

Una solución es que un agente de transferencia de mensaje (MTA) en una maquina ISP acepte correo electrónico para sus clientes y lo almacene en sus buzones en una maquina ISP. Puesto que este agente puede estar en línea todo el tiempo, el correo puede enviarse las 24 horas del día.

Desgraciadamente esta solución genera otro problema:

¿Cómo obtiene el usuario el correo electrónico del MTA del ISP?
La solución a este problema es crear protocolos que permitan que los MUA contacten al MTA (en la maquina ISP) y que el correo electrónico se copie desde el ISP al usuario. Tales protocolos son POP3 e IMAP.

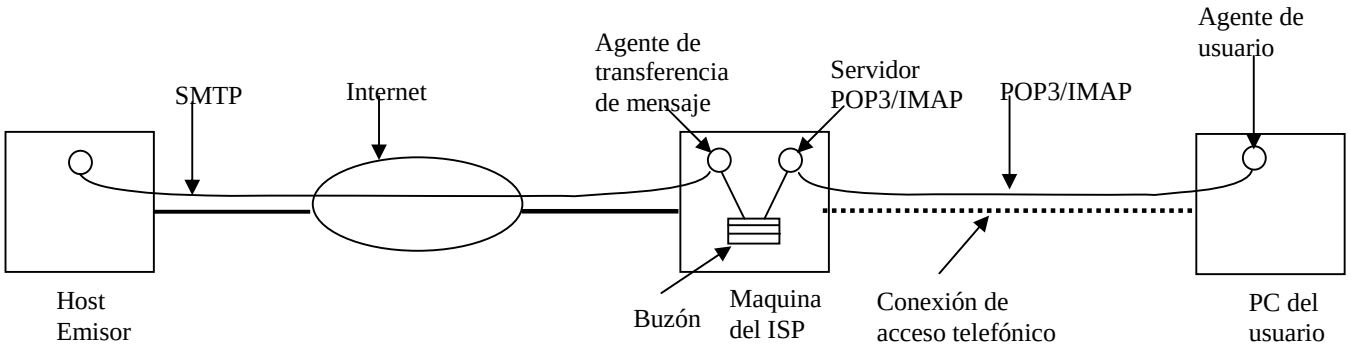

Figura 5: Lectura de correo cuando el receptor tiene una conexión de acceso telefónico con un ISP.

Postfix.

El MTA (Mail Transportation Agent) Postfix pretende ser rápido, fácil de administrar y seguro, a la vez lo suficientemente compatible con Sendmail como para que los usuarios existentes no se asusten.

Arquitectura.

Al contrario de Sendmail, que es un gestor de correo monolítico, en el diseño de Postfix se han disgregado los diversos tratamientos que se realizan sobre un mensaje a su paso por un Mail Transfer Agent (MTA), adjudicando cada tratamiento o grupo de tratamientos a un proceso independiente. El conjunto de todos estos procesos es Postfix.

Los procesos que conforman Postfix se comunican a través de sockets que se crean, por razones de seguridad, en un directorio de acceso restringido. La información que intercambian los diversos procesos es la mínima posible, limitándose en la mayoría de los casos a la referencia de la entrada en una cola y la relación de destinatarios, o a un simple identificador de estado.

La siguiente figura proporciona una visión global de los elementos que componen Postfix:

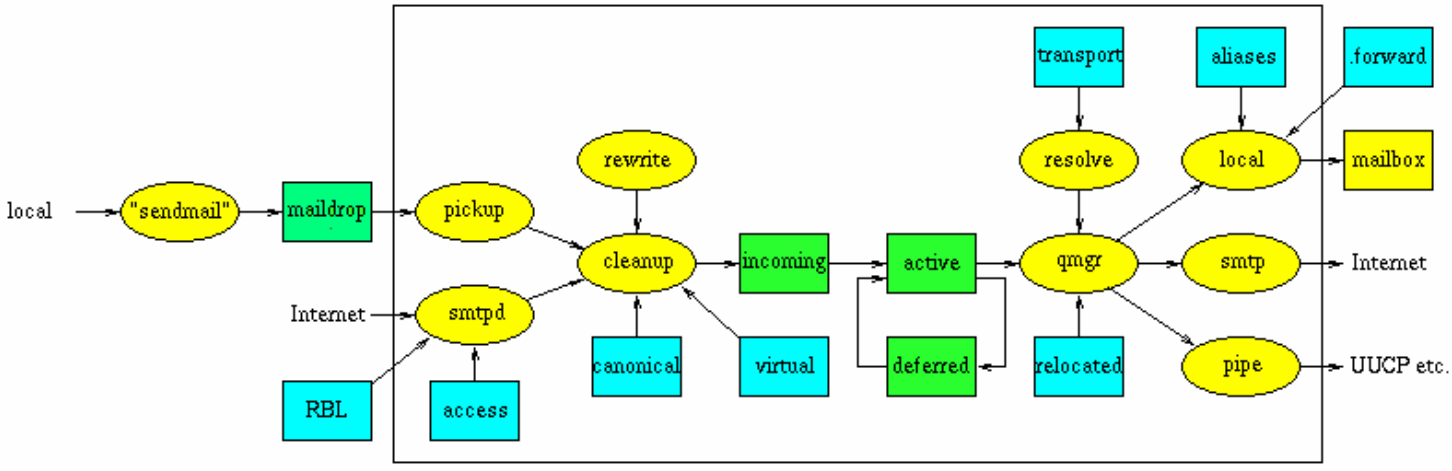

Figura 6: Visión Global de los elementos que componen Postfix.

Postfix basa su funcionamiento en cuatro colas: maildrop, incoming, active y deferred (cuadrados coloreados en verde).

- ✓ El correo que se genera de forma local se deposita en maildrop para su posterior proceso. El proceso pickup toma los mensajes que llegan a maildrop y los pasa a cleanup, que analiza las cabeceras de los mensajes y deposita éstos en la cola incoming.

- ✓ En la cola active se encuentran aquellos mensajes que están en fase de encaminamiento, y en deferred los mensajes que por diversas causas no se pueden encaminar o están pendientes de reintentar su encaminamiento.

- ✓ El proceso qmgr es el encargado de tratar los mensajes que llegan a la cola incoming, depositarlos en active y lanzar el proceso adecuado para su encaminamiento, como pueden ser local, smtp o pipe.

- ✓ El correo procedente de otros sistemas se atiende a través del proceso smtpd, utilizando el protocolo SMTP, pudiendo utilizar accesos a servidores de RBL o tablas internas para aplicar las políticas de acceso a cada mensaje entrante.

- ✓ Coloreadas de azul aparecen las tablas que, creadas por el administrador, sirven a los diferentes procesos para concretar el tratamiento que debe darse a cada mensaje. Se usan seis tablas: access, aliases, canonical, relocated, transport y virtual. Aunque no es obligatoria la existencia ni utilización de todas ellas.

- ✓ La tabla access permite definir una relación explícita de sistemas a los que se les deben aceptar o rechazar sus mensajes. La utiliza el proceso smptd.

- ✓ La tabla aliases, al igual que en Sendmail, define una serie de nombres alternativos a usuarios locales, y la consulta el proceso local.

- ✓ El proceso cleanup, mediante la tabla canonical establece relaciones entre nombres alternativos y nombres reales, ya sean usuarios locales o no.

- ✓ El proceso qmgr utiliza la tabla relocated para devolver los mensajes de usuarios que han cambiado de dirección: “User has moved to new-email”.

- ✓ Con la tabla transport, que es utilizada por el proceso trivial-rewrite, se define la política de encaminamiento por dominios, subdominios e incluso por dirección concreta de usuario.

- ✓ Para la gestión y soporte de dominios virtuales el proceso cleanup utiliza la tabla virtual. En ella se establecen las relaciones entre usuarios virtuales y reales, e incluso de dominios completos.

Todas estas tablas pueden usar alguno de los siguientes tipos de formato de base de datos:

- ✧ Fichero binario indexado (btree, hash, dbm, etc).
- ✧ Fichero de texto basado en expresiones regulares (regexp).
- ✧ Sistema externo de base de datos (NIS, LDAP, MySQL, etc).

Protocolo SMTP (Simple Mail Transfer Protocol).

Este protocolo es el estándar de Internet para el intercambio de correo electrónico. SMTP necesita que el sistema de transmisión ponga a su disposición un canal de comunicación fiable y con entrega ordenada de paquetes, con lo cual, el uso del protocolo TCP en la capa de transporte, es lo adecuado. Para que dos sistemas intercambien correo mediante el protocolo SMTP, no es necesario que exista una conexión interactiva, ya que este protocolo usa métodos de almacenamiento y reenvío de mensajes.

Son tres los protocolos que se aplican a un correo de esta clase. El termino SMTP es frecuentemente y erróneamente usado para referirse a la combinación del grupo de protocolos involucrados en el envío de correo electrónico. Esto porque los tres están estrechamente relacionados, pero estrictamente hablando SMTP es uno de los tres protocolos. Los tres protocolos son:

1. Un estándar para el intercambio de correo entre dos computadores (RFC 821), el cual especifica el protocolo usado para enviar correo entre "host" TCP/IP. Este estándar es SMTP.

2. Un estándar del formato del mensaje de correo, contenido en dos RFC:
    a. RFC 822 describe la sintaxis del campo de título o cabecera del correo electrónico y describe la interpretación del grupo de campos de la cabecera.
    b. RFC 1049 describe como un conjunto de otros tipos de documentos, que tengan texto ASCII, y que pueden ser usados en el cuerpo del correo electrónico. El nombre del protocolo oficial para este estándar es MAIL.

3. Un estándar para el "routing" de "mail" usando el sistema de nombres de dominio, descrito en RFC 974. El nombre oficial del protocolo para este estándar es DNS-MX.

Cyrus.

Es un sistema de mail diseñado principalmente para entornos de empresas o similares. Es altamente escalable y con un buen rendimiento. Implementa varios estándares, como IMAP y POP.

Razones por las cuales usar Cyrus:

- Altamente configurable.

- Sieve integrado.

- Permisos, quotas, etc implementadas en el propio sistema de Cyrus

- Modo de almacenar mails propio: Esto puede verse como una ventaja o una desventaja. Cyrus no es compatible con los métodos tradicionales de almacenar mails, como mailbox o maildir, si no que utiliza un formato propio. La ventaja de esto es que tiene mayor rendimiento accediendo a un gran número de mails. Ademas Cyrus también indexa los contenidos. Tenemos que tener además en cuenta que Postfix no podrá dejar los mails en el disco duro (en el mailbox), sino

que tendrá que pasarselos directamente a Cyrus mediante el protocolo LMTP que ambos implementan. De este modo Cyrus lo almacenará en su formato propio.

- Soporta LDAP: esto era uno de los requisitos iniciales, que el sistema IMAP se pudiera autentificar contra una base de datos LDAP.

- Soporta Dominios Virtuales.

- Cyrus IMAP (Internet Message Access Protocol) es desarrollado y mantenido por el Andrew Systems Group de la Carnegie Mellon University.

- A diferencia de otros servidores IMAP, Cyrus usa su propio método para almacenar el correo de los usuarios. Cada mensaje es almacenado en su propio fichero. El beneficio de usar ficheros separados es una mayor fiabilidad ya que sólo un mensaje se pierde en caso de error del sistema de ficheros. Los meta datos, tales como el estado de un mensaje (leído, etc.) se almacenan en una base de datos. Además, los mensajes son indexados para mejorar el rendimiento de Cyrus, especialmente con muchos usuarios e ingentes cantidades de mensajes. No hay nada tan rápido como el servidor IMAP Cyrus.

- Otra característica muy importante es que no son necesarias cuentas locales de Linux para cada usuario. Todos los usuarios son autenticados por el servidor IMAP. Esto lo convierte en una magnífica solución cuando se tiene una gran cantidad de usuarios.

- La administración es llevada a cabo mediante comandos especiales de IMAP. Esto le permite usar tanto la interfaz de línea de comandos como los interfaces Web.

Protocolo IMAP (Internet Message Access Protocol).

Es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet. IMAP tiene varias ventajas sobre POP, que es el otro protocolo empleado para obtener correo desde un servidor. Por ejemplo, es posible especificar en IMAP carpetas del lado servidor. Por otro lado, es más complejo que POP.

IMAP y POP3 (Post Office Protocol versión 3) son los dos protocolos que prevalecen en la obtención de correo electrónico. Todos los servidores y clientes de e-mail están virtualmente soportados por ambos, aunque en algunos casos hay algunas interfaces específicas del fabricante típicamente propietarias. Por ejemplo, mientras que los protocolos propietarios utilizados entre el cliente Microsoft Outlook y su servidor Microsoft Exchange Server o el cliente Lotus Notes de IBM y el servidor Domino, estos productos también soportan interoperabilidad con IMAP y POP3 con otros clientes y servidores.

Cyrus SASL (Simple Authentication And Security Layer).

Es un método para añadir soporte a la autenticación a protocolos basados en la conexión que ha sido estandarizado por la IETF (Internet Engineering Task Force). Se usa en servidores para manejar las peticiones de autenticación de los clientes.

Para ello, el protocolo incluye un comando para identificar y autenticar un usuario contra un servidor y para, opcionalmente negociar la protección de las subsiguientes interacciones del protocolo. Si se negocia su uso, una capa de seguridad es añadida entre el protocolo y la conexión. La librería SASL de Cyrus también usa la librería OpenSSL para cifrar los datos.

OpenSSL (Socket Secure Layer).

OPENSSL funciona como un algoritmo del tipo de clave pública asimétrica. En las claves públicas asimétricas el servidor crea un par de llaves que consiste en una llave/clave pública y una llave/clave privada. El servidor hace un requerimiento de llave pública a una entidad certificadora basada en la llave privada del servidor, la entidad certificadora modifica la llave de requerimiento de llave pública y le incrusta la llave pública de la entidad certificadora. Ahora el servidor quien hizo el requerimiento recibe la nueva y única llave publica que la entidad certificadora podrá verificar con su llave privada en el futuro.

El objetivo es cifrar la conexión entre el cliente (Browser) y el servidor (Web) con la llave pública única que siempre le envía el servidor al cliente. El servidor transmite la llave pública, poniéndola a disposición de cualquier browser o cliente que quiera conectarse al servidor por el puerto seguro 443.

Una vez que la llave publica ha sido recibida por el explorador / Browser o cliente este envía la llave publica del servidor a la entidad certificadora para su verificación y aprobación, solo la entidad certificadora podrá verificar con su llave privada la validez de la llave publica.

OpenSSL es una librería necesaria por SASL para la encriptación del flujo de datos. Es usado por casi todo el software opensource que necesita encriptación. La mayoría de las distribuciones Unix vienen con OpenSSL preinstalado.

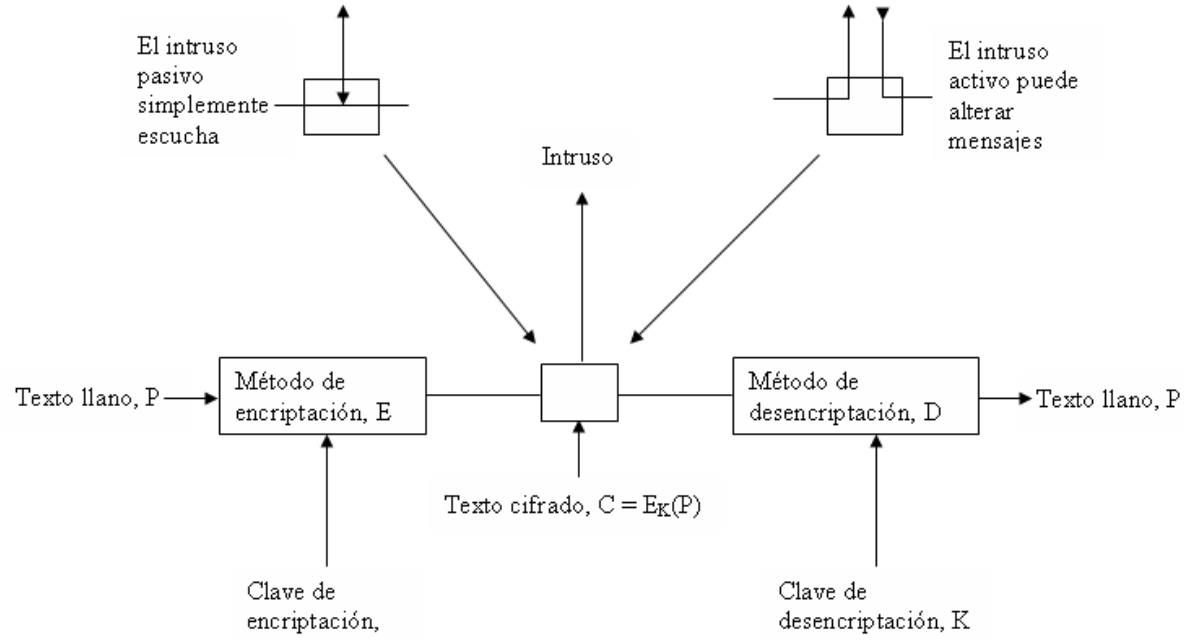

Figura 7: Modelo de encriptación (para un cifrado de clave simétrica porque utiliza la misma clave para encriptar y desencriptar).

Protocolo SSH (Secure Shell).

Es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X arrancado.

Además de la conexión a otras máquinas, SSH nos permite copiar datos de forma segura (tanto ficheros sueltos como simular sesiones FTP cifradas), gestionar claves RSA para no escribir claves al conectar a las máquinas y pasar los datos de cualquier otra aplicación por un canal seguro tunelizado mediante SSH.

SSH trabaja de forma similar a como se hace con telnet. La diferencia principal es que SSH usa técnicas de cifrado que hacen que la información que viaja por el medio de comunicación vaya de manera no legible y ninguna tercera persona pueda descubrir el usuario y contraseña de la conexión ni lo que se escribe durante toda la sesión; aunque es posible atacar este tipo de sistemas por medio de ataques de REPLAY y manipular así la información entre destinos.

Sieve.

Es un lenguaje que puede usarse para crear filtros de correo electrónico en el momento de la entrega final del correo. No está ligado a ningún sistema operativo o servidor de correo en particular. Requiere el uso de la especificación de mensajes del RFC 822.

El lenguaje es suficientemente potente para ser útil, pero está limitado de modo que permita la creación de sistemas de filtrados seguros en el lado del servidor. El objetivo es no permitir a los usuarios hacer nada más complejo (y peligroso) que escribir sencillos filtros de correo, además de facilitar editores basados en interfaces gráficas de usuario.

El lenguaje no permite definir bucles o funciones, ni tampoco proporciona variables, se supone que el uso del lenguaje tiene lugar al final de la entrega, cuando el mensaje se mueve a una cuenta accesible por el usuario. En aquellos sistemas donde el MTA (Mail Transport Agent) realiza la entrega final (como es tradicional en los sistemas UNIX), es razonable clasificar cuando el MTA deposita el correo en la cuenta del usuario.

Sin embargo, los filtros Sieve pueden ser usados por varios puntos finales de entrega del sistema de correo: por el servidor SMTP, por un servidor IMAP o POP que archive una o más cuentas de usuario, o por un cliente de correo (MUA, Mail User Agent) que actúe como gestor de las entregas (por ejemplo, un cliente POP o IMAP sin conexión).

Amavisd-New.

Es una herramienta de código abierto que sirve de interfaz entre un servidor de correo y un antivirus y otras formas de comprobar los contenidos. Aunque algunos antivirus proporcionan sus propios mecanismos para filtrar el correo en el servidor.

Amavisd-new ofrece ventajas con respecto al rendimiento en algunos entornos, y además proporciona un punto de configuración sencillo y neutral para gestionar el filtrado, tanto del spam como de los correos infectados con virus.

Es una aplicación basada en un script de Perl flexible y de alto rendimiento que se ejecuta como un servicio, con un proceso maestro y otro hijo. Actúa como un servidor SMTP, recibiendo el mensaje de correo del servidor SMTP (por ejemplo, Postfix, exim o qmail), procesándolo y enviándolo o devolviéndolo al servidor SMTP.

Soporta herramientas antispam como SpamAssasin, además de un amplio rango de antivirus comerciales y de código abierto. El popular antivirus Clam es soportado de tres formas diferentes:

1. El servicio clamd (mejor rendimiento).

2. El paquete Perl Mail::ClamAV (no tan bueno).

3. La opción de la línea de comandos (como respaldo cuando clamd no esté disponible, por ejemplo).

También soporta otros antivirus populares, incluidos F-Prot, Sophos, Grisoft's AVG, KasperskyLab AVP, Antivir, F-Secure, McAffe y Panda.

Pero tenga en cuenta la licencia del producto que escoja. Obviamente con Clamav no tendrá que pagarse ninguna, pero con alguna de las otras opciones puede que tenga que pagarse la licencia correspondiente para un servidor SMTP, que es significativamente superior que la licencia para un equipo.

Amavisd-new puede ser configurado para bloquear los ficheros adjuntos con las extensiones potencialmente peligrosas, como .exe, .bat y .vbs (particularmente peligrosas para los clientes Windows que puedan estar usando el servidor SMTP). Se pueden también especificar un amplio rango de decodificadores/descompresores para examinar los archivos comprimidos como .cpio, .rpm, .deb, .zoo, .tar, .gz y .bz2.

Teóricamente la aplicación puede soportar cualquier servidor SMTP, pero funciona mejor con los habituales, incluyendo Sendmail, exim y qmail, aunque con el que mejor funciona de todos es con Postfix, el cual permite reintroducirse el correo a sí mismo después de filtrar el contenido.

SpamAssassin normalmente identifica acertadamente entre un 95 y un 99% del spam, dependiendo del tipo de correo que se reciba. También incluye soporte para informar de mensajes de spam, automática o manualmente, a bases de datos como Vipul's Razor.

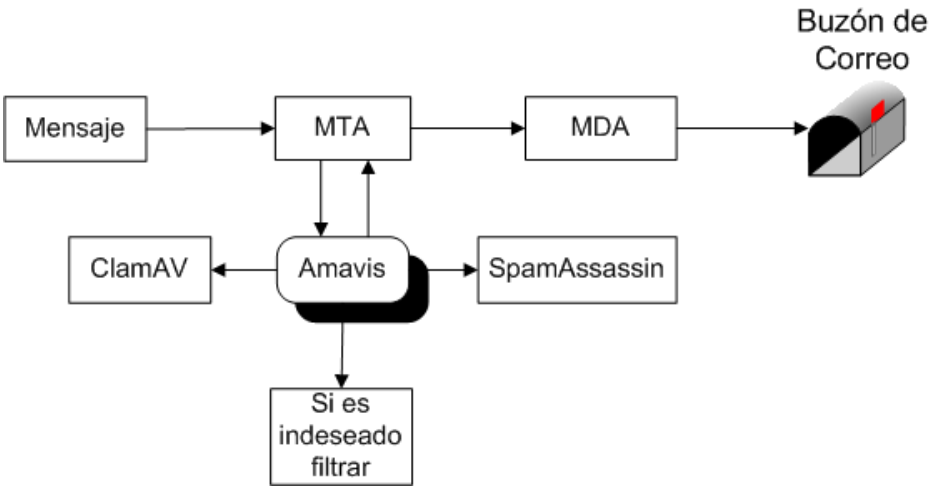

Figura 8: Filtro de correo electrónico.

Clam Antivirus.

ClamAV es una herramienta antivirus GPL para UNIX, el propósito principal de este software es la integración con los servidores de correo (escaneo de datos adjuntos). El paquete proporciona un servicio multihilo flexible y escalable, un analizador de línea de comandos y una utilidad para la actualización automática vía Internet.

Los programas están basados en una librería distribuida con el paquete Clam AntiVirus, el cual puede ser usado por su propio software. Y lo más importante, la base de datos se mantiene actualizada constantemente.

Otras características destacables son el soporte de firmas digitales en la actualización de la base de datos, el análisis durante el acceso bajo Linux y FreeBSD, la detección de más de 20000 virus, gusanos y troyanos, el soporte integrado para archivos comprimidos con Rar, Zip, Gzip y Bzip2 y formatos de correo Mbox, Maildir y ficheros crudos de correo.

Spamassassin.

SpamAssassin es una herramienta para inspeccionar correos electrónicos que permite determinar si se trata de un mensaje chatarra, mejor conocido como SPAM.

En este sentido Spamassasin es considerado un pre-procesador de correos, ya que la inspección es llevada acabo en el servidor de correos antes que el usuario descargue su correo, así permitiendo una pre-clasificación de mensajes antes de utilizar una herramienta en una PC(mozilla, etc.).

SpamAssassin utiliza varios criterios para determinar si un mensaje es SPAM:

- ✧ Inspección de "Headers": Los "Headers" o cabeceras de mensaje contienen información importante acerca del mensaje, como lo son procedencia y rutas de servidor, SpamAssassin inspecciona esta información para fines de detección.

- ✧ Análisis del Mensaje: El cuerpo y titulo del mensaje también son leídos por SpamAssassin, realizando búsquedas por palabras claves o estructuras que conforman un correo chatarra.

- ✧ Listas Negras: Actualmente, existen listas que enumeran servidores de correo conocidos como generadores de SPAM ("Open-Relays"), SpamAssassin consulta estas listas negras.

- ✧ Análisis probabilístico / bayesiano: Una vez definidas las reglas iniciales para detección, SpamAssassin utiliza análisis probabilístico para determinar similitudes entre mensajes entrantes y aquellos ya detectados como SPAM.

Listas "Hash" / Firmas de Correo: Debido a que un correo SPAM suele ser enviado a miles de personas a la vez, la estructura de cada mensaje es idéntica en todas sus instancias, así produciendo un "Hash" inequívoco. SpamAssasin consulta listas de "Hashes" sobre mensajes conocidos, como lo serian: Vipul's Razor , Pyzor y DCC.

Funcionamiento.

SpamAssassin funciona en modo texto: toma el mensaje de correo (cabeceras, cuerpo, todo) y busca determinados patrones. Por cada patrón que encuentra suma una determinada cantidad de puntos. Cuando los puntos superan un umbral el correo se marca como spam.

Tanto el umbral, como el valor de cada patrón, como los mismos patrones son configurables por cada usuario, que además puede añadir nuevos patrones a buscar. Los patrones existentes, que son muchos, van desde ver si el remitente tiene una dirección que empieza por un número hasta buscar frases en el cuerpo como "completamente gratis".

LMTP (Local Mail Transfer Protocol).

SMTP (Simple Mail Transfer Protocol) y sus extensiones ESMTP (SMTP Service Extensions), proporcionan los mecanismos necesarios para el trasporte de correo fiable y eficaz. El servidor bajo SMTP controlará las colas de envío de correo. LMTP actúa de forma que el servidor no tenga que manejar colas de correo, a modo de MDA (Mail Delivery Agent). Aunque LMTP puede utilizar extensiones que en un primer momento estarían definidas para su uso por ESMTP, nunca debería funcionar como receptor de escucha sobre el puerto 25.

SquirrelMail.

Es un interesante, extensible, funcional y robusto software para correo y que permite acceder al usuario a su correo electrónico desde el navegador de su preferencia.

SquirrelMail está escrito en PHP4, GPL (Licencia Pública General) y sigue el Standard HTML 4.0 para su presentación, haciéndolo compatible con la mayoría de servidores Web. SquirrelMail está diseñado para trabajar con plugins, lo cual hace más llevadera la tarea de agregar nuevas características entorno al núcleo de la aplicación.

Características.

- ✓ Gestión de carpetas.
- ✓ Internacionalización.
- ✓ Libro de direcciones personal y acceso a otros servicios de LDAP (muy útil si tienes LDAP montado en una organización). Permite hacer búsquedas de direcciones.
- ✓ Gestión de attachments.
- ✓ Servicio de búsqueda en e-mails.
- ✓ No necesita ninguna base de datos para funcionar (al contrario que muchos otros Webmails que necesitan MySQL o PostgreSQL).
- ✓ Interfaz de usuario fácil y potente.
- ✓ Arquitectura de plugins.
- ✓ Múltiples temas.

Configuración de las vistas de mensajes: número de mensajes visibles en pantalla, campos visibles, orden, cada cuanto tiempo comprueba si hay nuevos mensajes, etc.

Ventajas.

- ✓ Tiene muchas funciones interesantes. No tiene nada que envidiar a muchos clientes de correo. Además está escrito en PHP4 y es GPL, por lo que podes ampliarlo, modificarlo fácilmente y es totalmente gratis. Gracias a la arquitectura de plugins podrás añadir otros plugins para incluir nuevas funciones.
- ✓ Es muchísimo más estable que bastantes clientes de correo.
- ✓ Acceso a nuestro correo desde cualquier sitio, basta tener cualquier ordenador con conexión a Internet y un navegador.

Podes acceder al correo de forma segura (a través de SSL).

Instalación y Configuración.

Suponiendo que la instalación de Apache, PHP4 y IMAP ya funciona correctamente la instalación no puede ser más sencilla, bajas la última versión y la pones en un directorio que puede acceder vuestro servidor Web.

Cambie al directorio **/usr/share/squirrelmail/config/** y ejecute el guión de configuración que se encuentra en el interior:

```
$ cd /usr/share/squirrelmail/config/
$ ./conf.pl
```

Lo anterior le devolverá una interfaz de texto muy simple de utilizar, como se muestra a continuación:

```
SquirrelMail Configuration : Read: config.php (1.4.3)
---------------------------------------------------------
Main Menu --
1.  Organization Preferences
2.  Server Settings
3.  Folder Defaults
4.  General Options
5.  Themes
6.  Address Books (LDAP)
7.  Message of the Day (MOTD)
8.  Plugins
9.  Database


D.  Set pre-defined settings for specific IMAP servers

C.  Turn color on
S   Save data
Q   Quit

Command >>
```

Figura 9: Menú Principal SquirrelMail.

Ingrese hacia las preferencias de la organización y defina el nombre de la empresa, el logotipo y sus dimensiones, El mensaje en la barra de título de la ventana del navegador, el idioma a utilizar, URL y el título de la página principal del servidor de red.

```
SquirrelMail Configuration : Read: config.php (1.4.3)
---------------------------------------------------------
Organization Preferences
1.  Organization Name      : Razón_Social_de_su_empresa
2.  Organization Logo      : ../images/sm_logo.png
3.  Org. Logo Width/Height : (308/111)
4.  Organization Title     : Bienvenido al Webmail de Su_empresa.
5.  Signout Page           :
6.  Default Language       : es_ES
7.  Top Frame              : _top
8.  Provider link          : http://url_de_su_empresa/
9.  Provider name          : Nombre_de_su_emrpesa

R   Return to Main Menu
C.  Turn color on
S   Save data
Q   Quit
Command >>
```

Figura 10: Preferencias de Organización.

En las opciones de servidores defina solamente el dominio a utilizar. Si el servidor de correo va a coexistir en el mismo sistema con el servidor HTTP, no hará falta modificar más en esta sección. Si lo desea, puede especificar otro servidor SMTP e IMAP localizados en otro equipo.

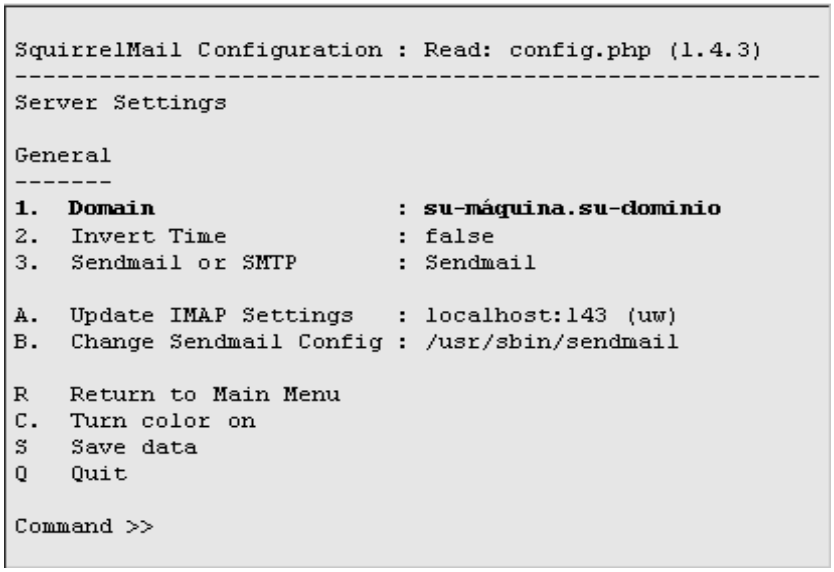

Figura 11: Configuración de Servidor.

En las opciones de las carpetas cambie Trash por Papelera, Sent por Enviados y Drafts por Borradores.

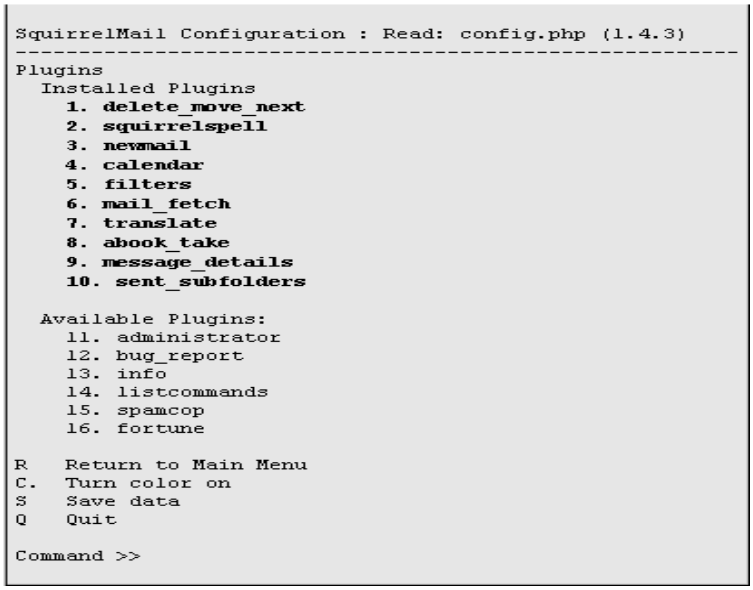

Figura 12: Plugins SquirrelMail.

Finalmente escoja y habilite las extensiones (plugins) que considere apropiados para sus necesidades.

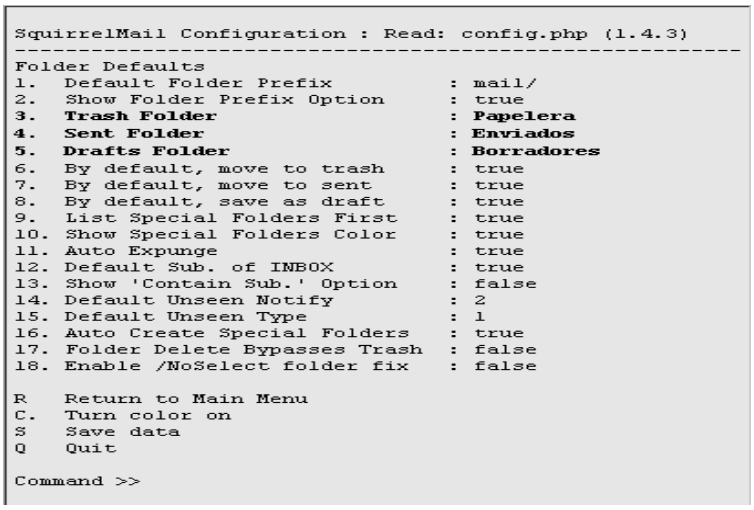

Figura 13: Directorios por defecto donde se almacenan los mensajes.

Guarde los cambios pulsando la tecla «S» y luego la tecla «Enter».

Finalizando la configuración.

Active, si no lo ha hecho aún, el servicio de IMAP. Si utiliza Red Hat Enterprise Linux 4.0, CentOS 4.0 o White Box Enterprise Linux 4.0, el paquete imap es reemplazado por dovecot, el cual funciona como otros servicios. Se debe editar el fichero /etc/dovecot.conf y asegurarse que estén habilitados el servicio de imap (de modo predefinido solo debe estar habilitado imap):

```
Protocols = imap pop3
```

El servicio se agrega al arranque del sistema y se inicializa del siguiente modo:

```
$ /sbin/chkconfig dovecot on
$ /sbin/service dovecot start
```

Reinicie o inicie el servicio de apache:

```
$ rcapache start
```

Acceda con el navegador de su preferencia hacia http://127.0.0.1/webmail/ o http://www.midominio.com/webmail/

PAM (Pluggable Authentication Module).

Linux utiliza PAM (Pluggable Authentication Modules) en el proceso de autenticación como una capa que media entre el usuario y la aplicación. Los módulos PAM están disponibles para todo el sistema, por lo que los puede solicitar cualquier aplicación.

Los administradores de sistemas y los programadores suelen restringir el acceso a ciertas partes del sistema o limitar el uso de ciertas funciones de una aplicación. Sin PAM, sería necesario adaptar las aplicaciones cada vez que se introdujera un nuevo mecanismo de autenticación, como LDAP o SAMBA. Sin embargo, este proceso lleva bastante tiempo y tiende a producir errores.

Una manera de evitar estos inconvenientes es separar las aplicaciones del mecanismo de autenticación y delegarlas en módulos gestionados centralmente. Cuando sea necesario utilizar un nuevo esquema de autenticación, bastará con adaptar o escribir un módulo PAM adecuado para que el programa en cuestión pueda utilizarlo. Todas las aplicaciones que utilicen un módulo PAM llamarán a un conjunto de funciones PAM, las cuales procesarán después la información en los distintos archivos de configuración y devolverán el resultado a la aplicación que ha realizado la llamada.

PAM no es un modelo de autenticación en sí, sino que se trata de un mecanismo que proporciona una interfaz entre las aplicaciones de usuario y diferentes métodos de autenticación, tratando de esta forma de solucionar uno de los problemas clásicos de la autenticación de usuarios: el hecho de que una vez que se ha definido e implantado cierto mecanismo en un entorno, es difícil cambiarlo.

Mediante PAM podemos comunicar a nuestras aplicaciones con los métodos de autenticación que deseemos de una forma transparente, lo que permite integrar las utilidades de un sistema Unix clásico (login, ftp, telnet...) con esquemas diferentes del habitual password: claves de un solo uso, biométricos, tarjetas inteligentes.

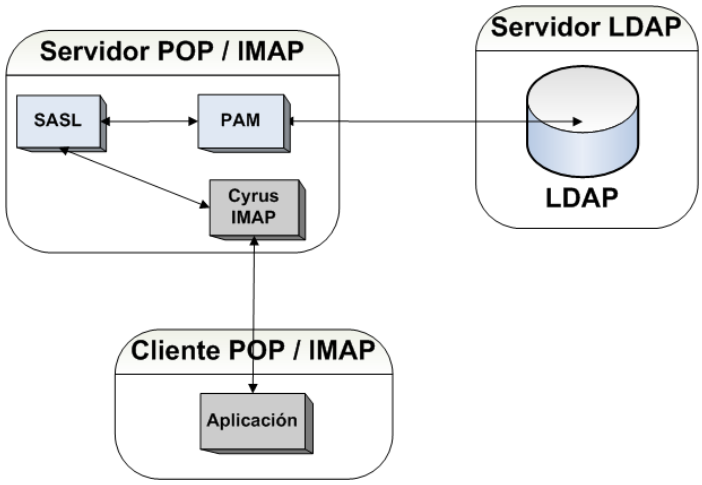

Figura 14: PAM como interfaz entre la aplicación de usuario y el mecanismo de autentificación LDAP.

Autenticación en el acceso al buzón.

Los protocolos de acceso al buzón como POP o IMAP siempre han utilizado autenticación de usuario para mantener la confidencialidad de los mensajes del usuario. Originalmente los datos de identificación usados por los servidores POP e IMAP coinciden con los que usa el sistema para autenticar a sus usuarios, normalmente los datos de /etc/passwd. También pueden usar los de alguna base de datos específica del servidor.

Las nuevas versiones ya están incluyendo otras opciones, por ejemplo el servidor cyrus IMAP utiliza el conjunto SASL y PAM. Con estas opciones los datos de identificación de los usuarios de correo pueden estar alojados en un servidor LDAP centralizado, y servir tanto para autenticar los envíos de correo como para el acceso al buzón del usuario.

Cuando una aplicación preparada para PAM inicia, se activa su comunicación con la API de PAM. Entre otras cosas esto fuerza la lectura del archivo de configuración: /etc/pam.conf. Alternativamente puede ser que se inicie la lectura de los archivos de configuración bajo /etc/pam.d/ (cuando existe un archivo de configuración correcto bajo este directorio, se ignora el archivo /etc/pam.conf).

Estructura de archivos de configuración PAM:

Cada línea dentro de un archivo de configuración PAM contiene un máximo de cuatro columnas:

<Tipo de módulo> <Indicador de control> <Vía al módulo> <Opciones>

Los módulos PAM se procesan en stacks. Los distintos tipos de módulos sirven a propósitos distintos, por ejemplo, un módulo comprueba la contraseña, otro verifica la ubicación desde la que se accede al sistema y otro lee los ajustes específicos del usuario.

PAM reconoce cuatro tipos de módulos diferentes:

- **auth**

El objetivo de este tipo de módulo es comprobar la autenticidad del usuario. Esto se hace tradicionalmente solicitando una contraseña, si bien también se puede conseguir con la ayuda de una tarjeta de chip o mediante biométrica (huellas digitales o exploración de retina).

- **account**

Los módulos de este tipo comprueban que el usuario tenga permiso general para utilizar el servicio solicitado. Por ejemplo, deberá realizarse esta comprobación para garantizar que nadie inicie sesión con el nombre de usuario de una cuenta caducada.

- **password**

El propósito de este tipo de módulo es permitir modificar un testigo de autenticación. En la mayoría de los casos, se trata de una contraseña.

- **sesión**

Los módulos de este tipo son responsables de gestionar y configurar sesiones de usuario. Estos módulos se inician antes y después de la autenticación para registrar intentos de inicio de sesión en los registros del sistema y para configurar el entorno específico del usuario (cuentas de correo, directorio personal, límites del sistema, etc.).

La segunda columna contiene indicadores de control para influir en el comportamiento de los módulos iniciados:

- **required**

Los módulos con este indicador se deben procesar correctamente antes de proceder con la autenticación. Si falla un módulo con el indicador required, se procesará el resto de módulos de este tipo antes de que el usuario reciba un aviso de que se ha producido un fallo durante el intento de autenticación.

- **requisite**

Los módulos con este indicador tienen que ser procesados correctamente, igual que los módulos con el indicador required. No obstante, si se produce un fallo en un módulo con este indicador, el usuario recibirá una notificación inmediata y no se procesarán más módulos. En caso de que no haya errores, se seguirá procesando el resto de los módulos, al igual que en el caso de los módulos con el indicador required. El indicador requisite se puede utilizar como un filtro simple con el objeto de comprobar el cumplimiento de determinadas condiciones necesarias para una correcta autenticación.

- **sufficient**

Si se procesa correctamente un modulo con este indicador, la aplicación que lo ha iniciado recibe inmediatamente una notificación de proceso correcto y no se procesa ningún otro módulo, siempre y cuando anteriormente no haya fallado la ejecución de ningún módulo con el indicador required. El fallo de un módulo con indicador sufficient no tiene consecuencias directas y los módulos siguientes se seguirán procesando según el orden correspondiente.

- **optional**

Que el proceso de un módulo con este indicador se lleve a cabo correctamente o haya errores no tiene consecuencias directas. Esta opción puede ser útil, por ejemplo, para módulos cuyo único cometido es mostrar un mensaje (por ejemplo informando al usuario acerca de la recepción de un mensaje de correo electrónico), sin realizar ninguna otra acción.

- **include**

Si se da este indicador, el archivo especificado como argumento se inserta en este lugar.
La vía al módulo no tiene por qué especificarse de forma explícita siempre que el módulo se encuentre en el directorio por defecto /lib/security (en todas las plataformas de 64 bits compatibles con SUSE Linux, el directorio es /lib64/security).

La cuarta columna puede contener una opción para el módulo, como debug (activa la depuración) o nullok (permite utilizar contraseñas vacías).

Configuración PAM para SSHD.

Tomemos la configuración PAM para sshd para demostrar la teoría con un ejemplo práctico de funcionamiento:

```
auth        include     common-auth
auth        required    pam_nologin.so
```

```
account     include     common-account
password    include     common-password
session     include     common-session
```

La configuración típica PAM de una aplicación (sshd en este caso) contiene instrucciones que hacen referencia a los archivos de configuración de cuatro tipos de módulos:

common-auth, common-account, common-password y common-session.

Estos cuatro archivos contienen la configuración predeterminada para cada tipo de módulo. Si se incluyen estos archivos en lugar de llamar a cada módulo por separado para cada aplicación PAM, se obtendrá automáticamente una configuración PAM actualizada cuando el administrador cambie los ajustes por defecto. Antiguamente, era necesario ajustar todos los archivos de configuración manualmente en todas las aplicaciones cuando se producían cambios en PAM o cuando se instalaba una aplicación nueva. Ahora, la configuración PAM se lleva a cabo mediante archivos de configuración centrales y todos los cambios se heredan automáticamente en la configuración PAM de cada servicio.

LDAP (LIGHTWEIGHT Directory Access Protocol).

El protocolo ligero de acceso al directorio (LDAP) es un conjunto de protocolos diseñados para acceder a los directorios de información y mantenerlos. Puede usarse LDAP con varios propósitos, como la gestión de usuarios, grupos, configuraciones del sistema y direcciones.

En un entorno de red es fundamental mantener la información importante estructurada y disponible rápidamente. Esto puede realizarse con un servicio de directorio que, como las páginas amarillas, mantenga la información disponible con un formato de búsqueda rápido y bien estructurado.

En una situación ideal, un servidor central mantiene los datos en un directorio y los distribuye a todos los clientes mediante un protocolo concreto. Los datos se estructuran de manera que permiten que una amplia gama de aplicaciones acceda a ellos.

De esta forma, no es necesario que cada herramienta de calendario o cliente de correo electrónico mantenga su propia base de datos. En lugar de ello, se puede acceder a un repositorio central. De ese modo se reduce notablemente el esfuerzo que requiere la administración de la información.

El uso de un protocolo abierto y estandarizado como LDAP asegura que todas las aplicaciones cliente puedan acceder a dicha información. Un directorio en este contexto es un tipo de base de datos optimizada para una lectura y búsqueda rápida y efectiva:

Para hacer posible varios accesos de lectura (simultáneos), el administrador ha limitado el acceso de escritura a un pequeño número de actualizaciones. Las bases de datos convencionales están optimizadas para aceptar el mayor volumen de datos posible en un corto espacio de tiempo.

Puesto que los accesos de escritura sólo se pueden ejecutar de una forma restringida, se emplea un servicio de directorio para administrar sobre todo información estática que no

cambia. Cuando se administran datos estáticos, las actualizaciones de los conjuntos de datos existentes son escasas.

Al trabajar con datos dinámicos, sobre todo en cuanto a los conjuntos de datos (cuentas bancarias o datos contables) se refiere, la coherencia de los datos es de suma importancia. Si se debe restar una cantidad de un lugar para sumarla en otro, ambas operaciones deben producirse a la vez, en una misma transacción, para asegurar el equilibro de los datos almacenados, las bases de datos admiten dichas transacciones.

Los directorios no. Las incoherencias de los datos por un corto período de tiempo son bastante aceptables en los directorios. Un servicio de directorio como LDAP no está diseñado para admitir actualizaciones complejas o mecanismos de consulta. Todas las aplicaciones que accedan a este servicio deben poder hacerlo de manera rápida y sencilla.

Han existido previamente muchos servicios de directorio y aún existen en Unix y fuera de él. Novell NDS, Microsoft ADS, Street Talk de Banyan y el estándar OSI X.500 son sólo algunos ejemplos. LDAP se diseñó en un principio como una variación simplificada de DAP, el protocolo de acceso al Directorio, que se desarrolló para acceder al X.500. El estándar X.500 regula la organización jerárquica de las entradas de directorio.

LDAP es una versión más sencilla de DAP. Sin perder la jerarquía de entradas de X.500, saca partido de las posibilidades de utilizar distintas plataformas de LDAP y ahorra recursos. El uso de TCP/IP hace mucho más sencillo establecer interfaces entre una aplicación de anclaje y el servicio LDAP.

LDAP, entretanto, ha evolucionado y se utiliza cada vez más como una solución autónoma sin la ayuda de X.500. LDAP es compatible con las referencias con LDAPv3, lo que permite disponer de bases de datos distribuidas.

LDAP no está limitado a la consulta de datos de los servidores X.500, tal y como se pensó en un principio. Hay un servidor de código abierto, slapd, que puede almacenar información de objetos en una base de datos local. También hay una extensión denominada slurpd, que es la responsable de replicar varios servidores LDAP.

Estructura de un árbol de directorios de LDAP.

Un directorio LDAP tiene una estructura de árbol. Todas las entradas (denominadas "objetos") del directorio tienen una posición definida en esta jerarquía. Esta jerarquía se denomina árbol de información del Directorio (DIT). La vía completa a la entrada deseada, que la identifica de forma clara, se llama nombre completo o DN. Un nodo sencillo junto con la vía a esta entrada se denomina nombre completo relativo o RDN.

Los objetos pueden asignarse generalmente a uno de dos tipos posibles:

- **Contenedor.**

Estos objetos pueden a su vez contener otros objetos. Tales clases de objetos son root (el elemento raíz del árbol de directorios, que no existe realmente), c (país), ou (unidad organizativa) y dc (componente de dominio). Este modelo es comparable con los directorios (carpetas) de un sistema de archivos.

- **Hoja.**

Estos objetos se encuentran en la parte final de una rama y no incluyen objetos subordinados. Algunos ejemplos serían person, InetOrgPerson o groupofNames. La parte superior de la jerarquía de directorios tiene un elemento raíz root. Puede contener a su vez c (país), dc (componente de dominio) ó o (organización) como elementos subordinados. Las relaciones dentro de un árbol de directorios LDAP se hacen más evidentes en el siguiente ejemplo:

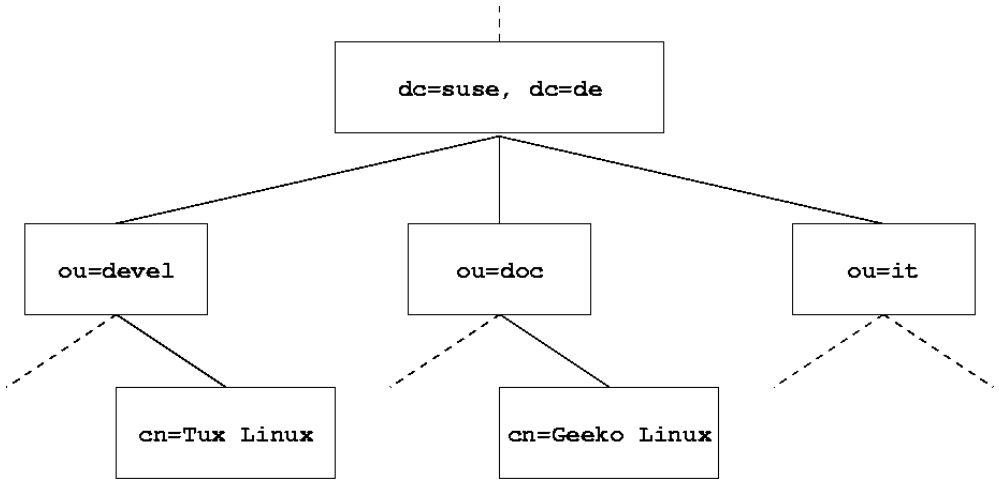

Figura 15: Estructura de un directorio LDAP.

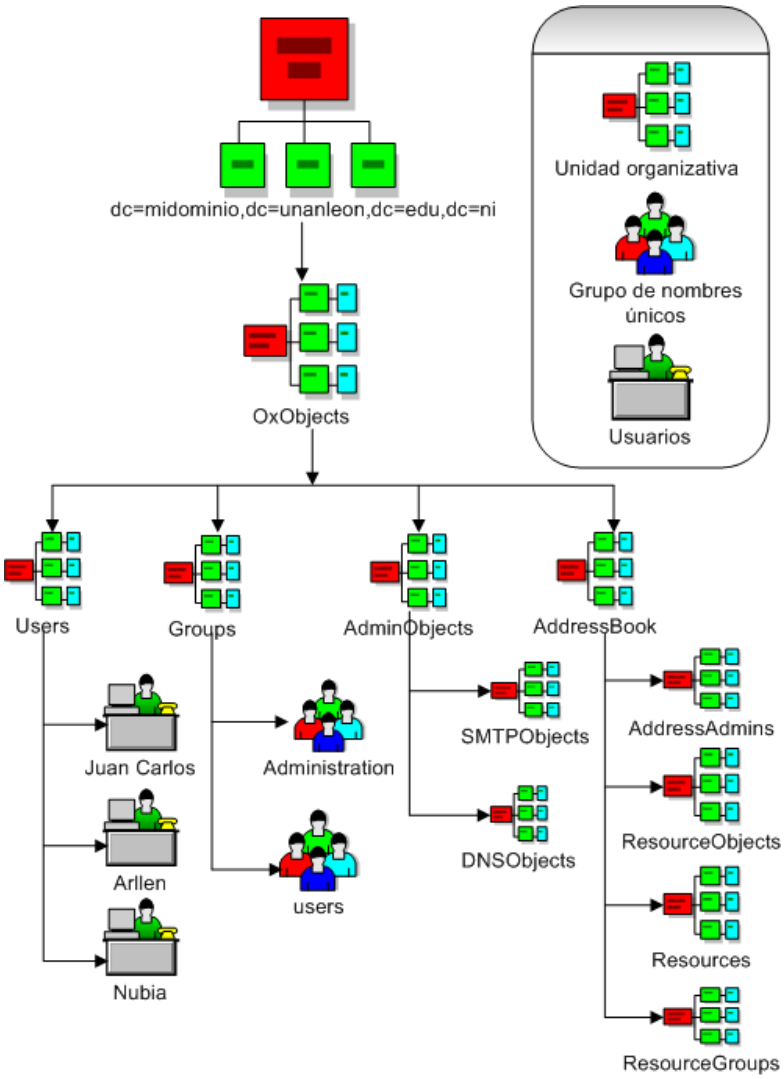

Figura 16: Estructura del árbol utilizado.

La determinación global de qué tipos de objetos deberían almacenarse en el DIT se realiza siguiendo un esquema. El tipo de objeto está determinado por la clase del objeto. Sirve para determinar qué atributos puede o debe asignarse el objeto en cuestión.

Un esquema, por tanto, debe contener definiciones de todas las clases y atributos del objeto utilizados en el escenario de la aplicación deseada. Hay unos cuantos esquemas comunes.

Sin embargo, es posible crear esquemas personalizados o usar varios esquemas que se complementen unos a otros si el entorno en el que debe operar el servidor LDAP lo necesita.

LDAP frente a NIS.

El administrador del sistema Unix normalmente utiliza el servicio NIS (Servicio de información de red) para la resolución de nombres y la distribución de datos por la red.
Los datos de configuración incluidos en los archivos de /etc y los directorios group, hosts, mail, netgroup, networks, passwd, printcap, protocols, rpc y services se distribuyen por los clientes por toda la red. Estos archivos pueden mantenerse sin gran esfuerzo porque son archivos de texto sencillos. La gestión de cantidades más grandes de datos, sin embargo, se vuelve cada vez más difícil debido a una estructura inexistente. NIS está diseñado únicamente para plataformas Unix, lo que significa que no se puede utilizar como herramienta de administración de datos centralizada en redes heterogéneas.

A diferencia de NIS, el servicio LDAP no está restringido a redes Unix puras. Los servidores Windows (a partir de la versión 2000) admiten LDAP como servicio de directorio. Novell también ofrece un servicio LDAP. Las tareas de aplicaciones mencionadas anteriormente también son compatibles en sistemas que no sean Unix.

El principio de LDAP puede aplicarse a cualquier estructura de datos que deba administrarse de manera centralizada. Algunos ejemplos de su aplicación son los siguientes:

a. Empleo como sustituto para el servicio NIS.

b. Encaminamiento de correo (postfix, sendmail).

c. Libretas de direcciones para clientes de correo como Mozilla, Evolution y Outlook.

d. Administración de descripciones de zona para un servidor de nombres BIND9.

e. Autenticación de usuarios con Samba en redes heterogéneas.

Esta lista puede ampliarse porque LDAP es extensible, al contrario que NIS. La estructura jerárquica claramente definida de los datos facilita la administración de grandes cantidades de ellos puesto que se puede buscar mejor.

PostgreSQL.

Es un servidor de base de datos relacional orientada a objetos de software libre, liberado bajo la licencia BSD (Berkeley Software Distribution). Es el motor de bases de datos de código abierto más potente del momento y en sus últimas versiones empieza a no tener que envidiarle nada a otras bases de datos comerciales.

Características.

a. **Alta concurrencia:** Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

b. **Cliente/Servidor:** PostgreSQL usa una arquitectura proceso por usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.
c. **Disparadores (triggers):** Un disparador o trigger se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:

   - El nombre del trigger o disparador.
   - El momento en que el disparador debe arrancar.
   - El evento del disparador deberá activarse sobre.
   - La tabla donde el disparador se activara.
   - La frecuencia de la ejecución.
   - La función que podría ser llamada.

   Entonces combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (triggers).

d. **Integridad Referencial:** PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

e. **Funciones:** Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientación a objetos o la programación funcional.

f. **Consistencia:** es la propiedad que asegura que sólo se empieza aquello que se puede acabar. Por lo tanto se ejecutan aquellas operaciones que no van a romper la reglas y directrices de integridad de la base de datos.

g. **Durabilidad:** es la propiedad que asegura que una vez realizada la operación, ésta persistirá y no se podrá deshacer aunque falle el sistema.

h. **Atomicidad (Indivisible):** es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias.

Mejoras en PostgreSQL.

Los bloqueos de tabla han sido sustituidos por el control de concurrencia multiversion, el cual permite a los accesos de solo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde pg_dump mientras la base de datos permanece disponible para consultas.

Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).
Se han añadido funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipo cadenas literales, conversión de tipos y entrada de enteros binarios y hexadecimales.

Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales. La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20-40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

Ventajas en PostgreSQL.

a. **Instalación ilimitada:** Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia.

b. **Mejor soporte que los proveedores comerciales:** Además de nuestras ofertas de soporte, tenemos una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribuir.

c. **Ahorros considerables en costos de operación:** Nuestro software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento.

d. **Estabilidad y confiabilidad legendarias:** En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad. Ni una sola vez. Simplemente funciona.

e. **Extensible:** El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

f. **Multiplataforma:** PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.

g. **Diseñado para ambientes de alto volumen:** PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mucho mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

JDBC (Java Data Base Connectivity).

Es una API (Application Programming Interface) que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.
El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos (URL) que pueden manejar.

Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la librería de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión, para ello provee un localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar cualquier tipo de tareas con la base de datos a la que tenga permiso: consultas, actualizaciones, creado modificado y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.

A la hora de conectarnos a una base de datos usando JDBC usamos un driver intermedio, que no es más que una clase ofrecida por el vendedor que implementa la interfaz Driver. Cuando se crea una instancia de una de estas clases Driver, esta se registra con el DriverManager (gestor de drivers) que es la encargada de decidir qué driver se ha de utilizar para acceder a tal base de datos.

Open-Xchange.

Es un sistema de mensajería diseñado para pequeñas y grandes empresas es una nueva plataforma de gran alcance de colaboración y trabajo en grupo. Aumenta la productividad a través de trabajo en equipo, reduce costos y tiene máxima flexibilidad técnica. Existen dos versiones de este producto: la versión comercial (software propietario) y la versión open-source (software libre).

Groupware se refiere a los programas informáticos que integran el trabajo de un proyecto con muchos usuarios concurrentes que se encuentran en diversas localizaciones o estaciones de trabajo, típicamente conectadas a través de la red Internet o de una intranet.

Flexible y accesible.

Open-Xchange es una plataforma web rentable construida con estándares abiertos de fuente optimizada para compañías con 5 a 5.000 empleados. Permite a sus empleados de todo el mundo comunicar e intercambiar rápida y eficientemente la información. Usando apenas un navegador, los empleados pueden tener acceso a todos sus e-mails así como su depósito de documentos, tareas, contactos, calendario, favoritos en cuestión de segundos, sin importar su localización física.

Funcionamiento Orientado.

Open-xchange es un substituto del alto rendimiento y bajo costo para la plataforma Exchange de Microsoft. Con una funcionalidad completa de una plataforma madura de la colaboración. OX maneja no solamente citas y tareas, si no que también el e-mail, los calendarios, los contactos, los proyectos, los documentos, búsqueda y foros.

Con OX, usted puede manejar la información usando los favoritos que se enlazan a una variedad amplia de objetos de datos, tales como e-mail, hojas de cálculo o presentaciones. Si necesita consolidar las comunicaciones de su empresa, OPEN-XCHANGE le ofrece todo.

OX permite que usted conecte con Microsoft Outlook y los dispositivos Palm usando los conectores. De acuerdo con las tecnologías más avanzadas le ofrece una seguridad de la mejor clase contra virus y Spam. La arquitectura abierta de OX le ofrece la flexibilidad de configurar el software para ajustarlo a su infraestructura y proteja su inversión a largo plazo.

OX esta testeado en diferentes escenarios y por miles de usuarios en todo el mundo y es la inspiración de centenares de programadores que hacen subir como la espuma toda la comunidad de software libre. Toda esta energía y entrega mejora continuamente el producto, asegurando todas las expectativas de desarrollo.

Open-Xchange Server soporta:

- Dispositivos SyncML.
- Cualquier navegador.
- Microsoft Outlook y Microsoft Outlook Express gracias al Outlook OXtender. Se trata de un software que permite que el outlook interactué con el servidor Open-Xchange Server como si fuera un Microsoft Exchange Server.
- WebDav interfaz (XML), LDAP, iCal, HTTP(S), SMTP, IMAP, POP3 y SyncML.

OXtenders e integración.

El desarrollo de Open-Xchange se fundamenta en estándares como WebDAV y XML. Contiene modulo anti-spam y una API para JAVA. Otras características del software OX son:

- ✓ E-mail, colaboración y mensajería instantánea.
- ✓ Vista de equipo y funcionalidades de calendario y gestión de proyecto.
- ✓ Soporte para el intercambio de datos con ERP, CRM y aplicaciones Microsoft Office.
- ✓ Diferentes niveles de permisos y propiedad.
- ✓ Colocación flexible de frames.

- ✓ Configuración de atajos de teclado individuales.
- ✓ Carpetas públicas.
- ✓ Creación de las plantillas de la vista.
- ✓ Apariencia personalizable.

Beneficios.

- ❖ Aumenta la productividad y disminuye el coste total de la propiedad.
- ❖ Una seguridad más robusta.
- ❖ Una comunicación más rápida.
- ❖ Incremento en la flexibilidad e interoperabilidad.
- ❖ Interfaz mucho más sencilla y amigable.

Módulos al Open-Xchange.

 **Portal.**

El módulo de entrada a Open-Xchange constituye un intento de poner a disposición del usuario un resumen de lo ocurrido durante los últimos días y el trabajo previsto para el día de hoy y los venideros. Incluye citas, tareas y correos electrónicos, ordenados por tipo de datos, todo en una única página, muy fácil de leer. Todas las alertas y cabeceras sirven como enlaces a documentos, adjuntos y datos. Muy útil al llegar por la mañana a la oficina.

 **Calendario.**

El módulo de calendario simplifica la coordinación de reuniones en grandes grupos de trabajo. El estado de libre/ocupado muestra la disponibilidad de todos los participantes así como los recursos, como salas de reuniones y conferencias o proyectores. Open-Xchange es capaz de calcular la ventana de disponibilidad más próxima para los miembros de tu equipo basándose en los parámetros que se elijan.

Todos los participantes, tanto si sin empleados como contactos, recibirán una invitación de manera automática si así se desea. Los miembros del equipo pueden aceptar o rechazar la petición de reunión. Usando la vista del equipo puede obtenerse una vista de todas las entradas del calendario de su equipo en un día en particular con la información relevante enlazada.

 **Contactos.**

El módulo de contactos de Open-Xchange es la solución a todos estos dilemas. Una libreta de direcciones global, contactos internos y externos relacionados con cada proyecto y contactos privados de acceso restringido. Todos los tipos de contactos pueden manejarse usando las categorías. Y, lo mejor de todo, los contactos pueden enlazarse basándose en la relación que tienen contigo.

 **Tareas.**

El módulo de tareas de Open-Xchange puede ayudarnos a gestionar un equipo, a tener una vista de las reuniones y los participantes necesarios para lograr un hito (del inglés, milestone). Nos permite tener acceso a un montón de listas muy útiles: listas de proyectos, listas de tareas, etc. Esto nos permitirá organizarnos y priorizar nuestro trabajo. Además, podremos enlazar o adjuntar documentos a las tareas.

 **Proyectos.**

Es una herramienta de gestión de proyectos que le permite a los miembros del equipo acceder a la información que necesitan para llevar a cabo sus tareas. Hoy en día, casi todas las empresas enfatizan muchísimo el trabajo en equipo. Un equipo bien gestionado es altamente productivo y ayuda a la empresa a mejorar su posición de mercado.

Pero a menudo es difícil organizar y gestionar un equipo. El software específico que promete automatizar la gestión de un proyecto a menudo no cumple las expectativas porque es complicado y difícil de aprender y usar. Lo que realmente se necesita es un módulo de proyectos que todo el mundo pueda usar porque haya sido diseñado para ser intuitivo.

Una herramienta de gestión de proyectos que permita a la gente acceder a la información que necesitan para su trabajo, tanto si esa información se encuentra en un correo electrónico, como en un evento del calendario, documento o mensaje de foro. Una plataforma que consigue que la información no se duplique y esté centralizada.

 **Documentos.**

Open-Xchange proporciona la parte más importante de una gestión de documentos, pues se concentra en aquellas características que nos permiten trabajar más rápido, incluso en grandes volúmenes de datos. Tanto si se trata de un control automático de versiones como si hay que bloquear un documento mientras se están editando, o si se trata de recuperar rápidamente un documento, Open-Xchange permite trabajar con los documentos usando herramientas comunes y de manera intuitiva. Además, nos permite acceder a documentos que necesitamos a través de Internet rápidamente y con seguridad, pues incluye un sistema de permisos.

 **Conocimiento.**

No todos en su empresa utilizarán las características avanzadas de estos módulos, pero es bueno saber que están ahí. Bases del conocimiento, marcadores globales, foros internos y boletines de noticias son algunos de los componentes avanzados de Open-Xchange, para que construya su sistema de colaboración empresaria de acuerdo a su compañía.

Open-Xchange puede ser un catalizador para crear una cultura de innovación, pues aúna las herramientas necesarias y las pone en las manos de los empleados, quienes las usarán para desarrollar, difundir y publicar sus ideas.

 **Favoritos.**

Permite visualizar las listas de sus favoritos de una manera ordenada a través de un árbol de directorio de fácil manejo. Esto le permitirá también editar, mover, borrar de una manera más sencilla.

 **Foro.**

Para ver y participar en el Foro institucional.

 **Tablero.**

Funciona como un “Periódico Mural Virtual” por área y de domino Público.

 **E-Mail.**

Esta opción le permitirá ingresar al su casilla de correo.

Software Libre.

El software Libre s aquel que puede ser distribuido, modificado, copiado y usado; por lo tanto, debe venir acompañado del código fuente para hacer efectivas las libertades que lo caracterizan. Es un tipo particular de software que le permite al usuario el ejercicio de cuatro libertades básicas:

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

Ventajas del Software Libre.

- ✓ **Escrutinio Público.**
  Al ser muchos las personas que tienen acceso al código fuente, eso lleva a un proceso de corrección de errores muy dinámico, no hace falta esperar que el proveedor del software saque una nueva versión.

- ✓ **Independencia del proveedor.**
  Al disponer del código fuente, cualquier persona puede continuar ofreciendo soporte, desarrollo u otro tipo de servicios para el software.
  No estamos supeditados a las condiciones del mercado de nuestro proveedor, es decir que si este se va del mercado porque no le conviene y discontinua el soporte, nosotros podemos contratar a otra persona.

- ✓ **Manejo de la Lengua.**
  Traducción: cualquier persona capacitada puede traducir y adaptar un software libre a cualquier lengua. Corrección ortográfica y gramatical: una vez traducido el software libre puede presentar errores de este tipo, los cuales pueden ser subsanados con mayor rapidez por una persona capacitada.

- ✓ **Mayor seguridad y privacidad.**
  Los sistemas de almacenamiento y recuperación de la información son públicos. Cualquier persona puede ver y entender como se almacenan los datos en un determinado formato o sistema. Existe una mayor dificultad para introducir código malicioso como ser: espía (p/ej. capturador de teclas), de control remoto, etc.

- ✓ **Garantía de continuidad.**
  El software libre puede seguir siendo usado aun después de que haya desaparecido la persona que lo elaboro, dado que cualquier técnico informático puede continuar desarrollándolo, mejorándolo o adaptándolo.

- ✓ **Ahorro en costos.**
  En cuanto a este tópico debemos distinguir cuatro grandes costos: de adquisición, de implantación (este a su vez se compone de costos de migración y de instalación), de soporte o mantenimiento, y de interoperabilidad. El software libre principalmente disminuye el costo de adquisición ya que al otorgar la libertad de distribuir copias la puedo ejercer con la compra de una sola licencia y no con tantas como computadoras posea (como sucede en la mayoría de los casos de software propietario). Cabe aclarar que también hay una disminución significativa en el costo de soporte, no ocurriendo lo mismo con los costos de implantación y de interoperatividad.

Desventajas del Software Libre.

Si observamos la situación actual, es decir la existencia mayoritaria de Software Propietario, tenemos:

- ✓ Dificultad en el intercambio de archivos: esto se da mayormente en los documentos de texto (generalmente creados con Microsoft Word), ya que si los queremos abrir con un Software Libre (p/ ej. Open Office o LaTeX) nos da error o

se pierden datos. Pero esta claro que si Microsoft Word creara sus documentos con un formato abierto (o público) esto no sucedería.

- ✓ Mayores costos de implantación e interoperabilidad: dado que el software constituye "algo nuevo", ello supone afrontar un costo de aprendizaje, de instalación, de migración, de interoperabilidad, etc., cuya cuantía puede verse disminuida por: mayor facilidad en las instalaciones y/o en el uso, uso de emuladores (p/ej. Si el usuario utiliza Microsoft Windows, la solución seria instalar alguna distribución de GNU/Linux y luego un emulador de Windows, como Wine, VMWare. Terminal X, Win4Lin). Vale aclarar que el costo de migración esta referido al software, ya que en lo que hace a Hardware generalmente el Software Libre no posee mayores requerimientos que el Software Propietario.

VIII. METODOLOGÍA.

1. Diseño metodológico.

   - ✓ Recopilación de información.
   - ✓ Análisis de la estructura de la red.
   - ✓ Instalación de programas (SO, servicios de red, etc)
   - ✓ Pruebas experimentales.

2. Recursos a emplear.

   - ✓ Sistema Operativo Linux: Distribución Suse 10.2.
   - ✓ Paquetería software (Open-Xchange, Postfix, Cyrus, LDAP, etc)

3. Hardware.

   - ✓ Servidor HP NetServer Lc 2000 U3 con Hardware interno:
     - ❖ Disco duro SHA 18 GB.
     - ❖ Disco duro SHB 18 GB.
     - ❖ Memoria RAM 512 MB.
     - ❖ Procesador 2.55 GHz.

4. Hardware adicional.

   - ✓ 2 interfaces de red Ethernet 10/100 Mbps con conector RJ45.

5. Personas.

   - ✓ Juan Carlos Bordas Montoya.
   - ✓ Arllen Javier Díaz Cáceres.
   - ✓ Nubia Consuelo Espinoza García.

IX. CONCLUSIONES.

Hemos cumplido exitosamente todos los objetivos propuestos en este trabajo monográfico. A continuación enumeramos nuestras conclusiones.

- ✧ Habilitamos la herramienta Open-Xchange con la cual se podrá revisar correos electrónicos, administrar contactos, realizar tareas, entre otros.

- ✧ Proporcionamos una documentación amplia de nuestro trabajo para que sirva de guía para otros proyectos.

- ✧ Configuramos todos los servicios básicos de red como son DNS, Apache, Tomcat, Open-Xchange, LDAP, Cyrus, Postfix.

- ✧ Constatamos que los sistemas GPLs son una muy buena alternativa ya que:

    - ➢ Son económicos respecto a los sistemas que tienen licencia de pago.
    - ➢ Cuentan con mejores mecanismos de seguridad.
    - ➢ Son de código abierto para realizarse mejoras y ajustarlo a nuestras necesidades.

Habilitamos herramientas administrativas vía Web para proporcionar a los administradores una alternativa más atractiva y sencilla de utilizar, además que son útiles para configuraciones posteriores de los servidores DNS, Web, Correo electrónico, LDAP, Open-Xchange, PostgreSQL.

X. RECOMENDACIONES.

Hacemos las siguientes recomendaciones:

- ✓ El servidor de correo debe estar separado en un solo PC, debido a que necesita mayor espacio en disco por la cantidad de cuentas de usuarios que hay que crear y por seguridad, cuando el número de usuarios es considerablemente grande.

- ✓ Que el sistema colaborativo sea usado a nivel de intranet debido a que las velocidades de los distintos medios de transmisión del país son muy lentas.

- ✓ Se debe de configurar un servidor DNS esclavo para utilizarlo de respaldo por si el DNS primario deja de prestar servicios.

- ✓ Utilizar hosts virtuales en el servidor Web Apache, esto con el objetivo de ahorrar esfuerzos administrativos y gastos de hardware.


ANEXOS

Archivos de Configuración.

DNS.

```
options {
        directory "/var/lib/named";
        dump-file "/var/log/named_dump.db";
        statistics-file "/var/log/named.stats";
        forwarders { 192.168.151.1; 192.168.151.8; };
        forward first;
        #listen-on port 53 { 127.0.0.1; };
        #listen-on-v6 { any; };
        #query-source address * port 53;
        #transfer-source * port 53;
        #notify-source * port 53;
        #allow-query { 127.0.0.1; };
        notify no;
};

zone "." in {
        type hint;
        file "root.hint";
};

zone "localhost" in {
        type master;
        file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
        type master;
        file "127.0.0.zone";
};

zone "midominio.unanleon.edu.ni" in {
        type master;
        file "dbmidominio";
};

zone "151.168.192.in-addr.arpa" in {
        type master;
        file "db151.168.192";
};
```

Figura 17: /etc/named.conf

Se puede dividir de forma somera en dos áreas. Una es la sección options para los ajustes generales, y la otra consiste en entradas zone para los dominios individuales. La sección de registro y las entradas de acl (lista de control de acceso) son opcionales. Las líneas de comentario comienzan con el signo almohadilla # o con dos barras //.

seccion option:

- ✓ **directory "nombre de archivo";**
  Especifique el directorio en el que BIND puede encontrar los archivos que º contienen los datos de la zona. Normalmente, se trata de /var/lib/named.
- ✓ **forwarders { ip-address; };**
  Especifica los servidores de nombres (la mayoría del proveedor) a los que se deberían remitir las peticiones DNS si no pueden resolverse directamente.

- ✓ **forward first;**
  Hace que las peticiones DNS se remitan antes de que se realice un intento para resolverlas mediante los servidores de nombres raíz. En lugar de forward first, se puede escribir forward only para remitir todas las peticiones y no enviar ninguna a los servidores de nombres raíz. Esto tiene sentido para las configuraciones de cortafuegos.

- ✓ **listen-on port 53 { 127.0.0.1; ip-address; };**
  Indica a BIND en qué interfaces de red y puerto se van a aceptar las consultas de clientes. No se tiene que especificar port 53 explícitamente, porque 53 es el puerto por defecto. Introduzca 127.0.0.1 para permitir peticiones desde el host local. Si omite esta entrada completamente, se usarán por defecto todas las interfaces.

- ✓ **listen-on-v6 port 53 {any; };**
  Indica a BIND en qué puerto debería escuchar para recibir las peticiones de cliente de IPv6. La única alternativa a any es none. Por lo que respecta a IPv6, el servidor sólo acepta direcciones comodín.

- ✓ **query-source address * port 53;**
  Esta entrada es necesaria si un cortafuego está bloqueando las peticiones de DNS salientes. Le indica a BIND que publique las peticiones externamente desde el puerto 53 y no desde los puertos superiores a 1024.

- ✓ **query-source-v6 address * port 53;**
  Indica a BIND qué puerto usar para las consultas de IPv6.

- ✓ **allow-query { 127.0.0.1; red; };**
  Define las redes desde las que los clientes pueden publicar las peticiones de DNS. Sustituya red por la dirección como, por ejemplo, 192.168.1/24. El /24 al final es una expresión abreviada para la máscara de red, en este caso, 255.255.255.0.

- ✓ **allow-transfer ! *;;**
  Controla qué hosts pueden solicitar transferencias de zona. En el ejemplo, se deniegan completamente tales peticiones mediante ! *. Sin esta entrada, las transferencias de zona pueden solicitarse desde cualquier parte sin restricciones.

- ✓ **statistics-interval 0;**
  Si esta entrada no está, BIND genera varias líneas de información estadística por hora en /var/log/messages. Defína ek valor 0 para suprimir estas estadísticas completamente, o bien establezca un intervalo en minutos.

- ✓ **cleaning-interval 720;**
  Esta opción define con qué frecuencia borra BIND su caché. Cada vez que ocurra, se activará una entrada en /var/log/messages. La especificación del tiempo se realiza en minutos. El valor por defecto es 60 minutos.

Entradas de Zonas:

Ejemplo:

```
zone "midominio.unanleon.edu.ni" in {
    type master;
    file "mi-dominio.zone";
    notify no;
};
```

Figura 18: Parte del archivo /etc/named.conf

Después de zone, especifique el nombre del dominio que se va a administrar, seguido por 'in' y una serie de opciones relevantes entre llaves

Opciones de zona:

- ✓ **type master;**
  Al especificar master, se indica a BIND que la zona está gestionada por el servidor de nombres local. De esta forma se asume que se ha creado un archivo de zona con el formato correcto.

- ✓ **type slave;**
  Esta zona se transfiere desde otro servidor de nombres. Debe usarse junto con masters.

- ✓ **type hint;**
  La zona . del tipo hint se utiliza para indicar los servidores de nombres raíz. Es una definición de zona que no es necesario modificar.

- ✓ **file "dbmidominio";**
  Esta entrada indica el archivo que contiene los datos de zona para el dominio. En caso de un esclavo no hace falta que el archivo exista, ya que se toma de otro servidor de nombres. Para separar los archivos esclavos de los principales, utilice slave como directorio de los archivos esclavos.

- ✓ **masters { dirección-ip-servidor; };**
  Esta entrada sólo es necesaria para las zonas esclavas. Indica desde qué servidor de nombres se debe transferir el archivo de zona.

- ✓ **allow-update {! *; };**
  Esta opción regula el acceso de escritura externo, lo que permitirá a los clientes crear su propia entrada de DNS (algo que normalmente no debe hacerse por motivos de seguridad). Sin esta entrada, las actualizaciones de zona están prohibidas. La entrada mencionada anteriormente obtiene los mismos resultados porque ! * prohíbe igualmente cualquier actividad.

Archivos de Zona:

Son necesarios dos tipos de archivos de zona. Uno asigna direcciones IP a nombres de host y el otro hace lo contrario: ofrece un nombre de host para una dirección IP.

```
1   $TTL 1W
2   @ IN SOA dns1.midominio.unanleon.edu.ni. arlen.midominio.unanleon.edu.ni.(
3                           42          ; serial (d. adams)
4                           2D          ; refresh
5                           4H          ; retry
6                           6W          ; expiry
7                           1W )        ; minimum
8
9                   IN  NS      dns1.midominio.unanleon.edu.ni.
10                  IN  MX 10   mail.midominio.unanleon.edu.ni.
11
12                  IN  A       192.168.151.167
13  dns1            IN  A       192.168.151.167
14  mail            IN  A       192.168.151.167
15  web             IN  A       192.168.151.167
16  www             IN  CNAME   mail
17  ldapserver      IN  A       192.168.151.167
```

Figura 19: /var/lib/named/dbmidominio

Línea 1.
$TTL define la duración por defecto que debería aplicarse a todas las entradas del archivo. En este ejemplo, las entradas son válidas durante un periodo de dos días (2 D).

Línea 2.
Aquí comienza la parte del registro de control SOA (Inicio de autoridad):

- ✓ En primer lugar figura el nombre del dominio que administrar midominio.unanleon.edu.ni. termina en punto porque, de lo contrario, se añadiría la zona otra vez. Una alternativa sería escribir el símbolo @, en cuyo caso la zona se extraería de la entrada correspondiente en /etc/named.conf.

- ✓ Detrás de IN SOA se encuentra el servidor de nombres que actuará como principal en esta zona. En este caso, el nombre se amplía de dns1 a dns1.midominio.unanleon.edu.ni ya que no termina en punto.

- ✓ A continuación aparece la dirección de correo electrónico de la persona que se encarga de este servidor de nombres. Como el símbolo @ ya tiene un significado especial, se reemplaza por un punto. Debe incluirse un punto al final para impedir que la zona se añada.

- ✓ El paréntesis de apertura (incluye todas las líneas que haya en el registro SOA hasta el paréntesis de cierre).

Línea 3.
El número de serie es un número al azar que debe aumentarse después de cada modificación del archivo. Es necesario para informar a los servidores de nombres

secundarios (servidores esclavos) acerca de los cambios. El formato más común para indicarlo es mediante una cifra de 10 dígitos formada por la fecha y el número de orden en la forma AAAAMMDDNN.

Línea 4.
La frecuencia de actualización especifica la frecuencia con la que los servidores de nombres secundarios verifican el número de serie de la zona. En este caso, un día.

Línea 5.
La frecuencia de reintento especifica después de cuánto tiempo el servidor de nombres secundario intenta conectar nuevamente con el servidor primario en caso de error. En este caso son dos horas.

Línea 6.
La hora de caducidad especifica el tiempo transcurrido después del cual el servidor de nombres secundario debe desechar los datos almacenados en caché si no se ha podido restablecer el contacto con el servidor primario. En este caso es una semana.

Línea 7.
La última entrada del registro SOA especifica el tiempo de vida (TTL) de almacenamiento en caché negativo; es decir, el tiempo que los resultados de las consultas DNS sin resolver de otros servidores pueden almacenarse en caché.

Línea 9.
IN NS especifica el servidor de nombres responsable de este dominio. En este caso dns1 se vuelve a convertir en dns1.midominio.unanleon.edu.ni porque no termina en punto. Puede haber varias líneas de este tipo, una para el servidor de nombres primario y otra para cada servidor de nombres secundario. Si la variable notify no está definida en no en /etc/named.conf, se informará a todos los servidores de nombres aquí mencionados de los cambios en los datos de zona.

Línea 10.
El registro MX indica el servidor de correo que acepta, procesa y remite los mensajes de correo electrónico al dominio midominio.unanleon.edu.ni. En este ejemplo, se trata del mail.midominio.unanleon.edu.ni. El número situado delante del nombre del host se corresponde con el valor de preferencia. Si existen varias entradas MX, primero se utiliza el servidor de correo con el valor de preferencia más bajo y, si la entrega del correo a este servidor no se produce, se hará un intento con el valor inmediatamente superior.

Líneas 12 a 15.
Se corresponden con los registros de direcciones reales en los que una o varias direcciones IP se asignan a nombres de host. Todos los nombres aparecen sin un punto porque no incluyen el dominio. Allí donde la dirección del host es de tipo tradicional (Ipv4), el registro aparece marcado con una A. Si la dirección es una dirección IPv6, la entrada aparece marcada con A6. El testigo anterior para las direcciones IPv6 era AAAA, que ahora está obsoleto.

Línea 16.
Se puede utilizar el alias www para acceder a Web (CNAME significa nombre canónico).

Resolución Inversa.

Para la resolución inversa de direcciones IP en nombres de host, se utiliza el seudo-dominio in-addr.arpa. Se añadirá al final de la parte correspondiente a la red de la dirección escrita en orden inverso. De tal forma que 192.168.151 se convierte en 151.168.192.in-addr.arpa.

```
1   $TTL 1W
2   @ IN SOA dns1.midominio.unanleon.edu.ni. arlen.midominio.unanleon.edu.ni.(
3                           42          ; serial (d. adams)
4                           2D          ; refresh
5                           4H          ; retry
6                           6W          ; expiry
7                           1W )        ; minimum
8
9           IN NS       dns1.midominio.unanleon.edu.ni.
10          IN MX   10  mail.midominio.unanleon.edu.ni.
11
12  167     IN PTR      midominio.unanleon.edu.ni.
13  167     IN PTR      dns1.midominio.unanleon.edu.ni.
14  167     IN PTR      mail.midominio.unanleon.edu.ni.
15  167     IN PTR      web.midominio.unanleon.edu.ni.
16  167     IN PTR      ldapserver.midominio.unanleon.edu.ni.
```

Figura 20: /var/lib/named/db151.168.192

Como se puede apreciar todo el comienzo es igual al anterior por lo que solo se explicara la parte que esta diferente al anterior.

Se hace uso de registros de puntero (PTR) que llevan a las direcciones IP de los hosts correspondientes. Al principio de la línea sólo se introduce la última parte de la dirección IP, sin el punto al final. Al añadir la zona al final (sin .in-addr.arpa) dará como resultado la dirección IP completa en orden inverso.

```
domain midominio.unanleon.edu.ni
search midominio.unanleon.edu.ni
nameserver 192.168.151.167
nameserver 192.168.151.1
nameserver 192.168.151.8
```

Figura 21: /etc/resolv.conf

domain: el parámetro domain sirve para no tener que introducir todo el nombre del dominio cuando queremos resolver el nombre de un host.

search: el parámetro search es un patrón de búsqueda donde se indica los dominios en donde buscar la información referente a un host determinado.

nameserver: el parámetro nameserver sirve para especificar quienes son nuestros servidores de nombres primarios y secundarios.

Correo Electrónico.

```
START {
  recover       cmd="ctl_cyrusdb -r"
  idled         cmd="idled"
}

# UNIX sockets start with a slash and are put into /var/lib/imap/socket
SERVICES {
  imap          cmd="imapd" listen="imap" prefork=0
  pop3          cmd="pop3d" listen="pop3" prefork=0
  sieve         cmd="timsieved" listen="127.0.0.1:sieve" prefork=0

# at least one LMTP is required for delivery
  lmtpunix      cmd="lmtpd" listen="/var/lib/imap/socket/lmtp" prefork=0
}

EVENTS {
  checkpoint    cmd="ctl_cyrusdb -c" period=30
  delprune      cmd="cyr_expire -E 3" at=0400
  tlsprune      cmd="tls_prune" at=0400
}
```

Figura 22: /etc/cyrus.conf

Este fichero de configuración consta de tres partes claramente diferenciadas:

START: Esta sección lista los scripts que se ejecutarán antes de que se arranquen los servicios. Su uso más característico es inicializar las bases de datos y lanzar los servicios de larga ejecución.

SERVICES: Esta sección es el corazón del fichero /etc/cyrus.conf, pues describe los procesos que deberán lanzarse para atender las conexiones que los clientes hagan a ciertos sockets, bien sean tipo TCP o UNIX.

EVENTS: Esta sección lista los procesos que deberían ejecutarse a intervalos específicos. Típicamente se usa para llevar a cabo tareas programadas de limpieza y mantenimiento.

En estos momentos entra en escena una cuestión importante en la configuración: usar sockets TCP o sockets UNIX. En la configuración que se acaba de presentar se ha optado por la segunda opción debido a que se considera que van a ejecutarse todos los servicios en la misma máquina.

Por tal razón está esta línea:

```
lmtpunix      cmd="lmtpd" listen="/var/lib/imap/socket/lmtp" prefork=0
```

Este servicio tiene que estar habilitado para el trasporte de correo. En un caso como éste, muy habitual, es mejor usar sockets UNIX debido, principalmente, al mejor rendimiento que ofrecen y a que simplifican la configuración en general. En cambio, en un

contexto donde los servidores Postfix y Cyrus IMAP estén en máquinas diferentes, será necesario usar sockets TCP.

```
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)

smtp               inet  n    -      n      -      -      smtpd

#submission     inet  n    -      n      -      -      smtpd
#          -o smtpd_etrn_restrictions=reject
#          -o smtpd_client_restrictions=permit_sasl_authenticated,reject

#smtps             inet  n    -      n      -      -      smtpd -o

smtpd_tls_wrappermode=yes
#  -o smtpd_tls_wrappermode=yes -o smtpd_sasl_auth_enable=yes

#submission      inet  n    -      n      -      -      smtpd
#  -o smtpd_etrn_restrictions=reject
#  -o smtpd_enforce_tls=yes -o smtpd_sasl_auth_enable=yes

#628            inet  n       -       n       -       -      qmqpd
pickup          fifo  n       -       n       60      1      pickup
cleanup         unix  n       -       n       -       0      cleanup
qmgr            fifo  n       -       n       300     1      qmgr
#qmgr           fifo  n       -       n       300     1      oqmgr
#tlsmgr         unix  -       -       n       1000?   1      tlsmgr
Rewrite         unix  -       -       n       -       -      trivial-rewrite
bounce          unix  -       -       n       -       0      bounce
defer           unix  -       -       n       -       0      bounce
trace           unix  -       -       n       -       0      bounce
verify          unix  -       -       n       -       1      verify
flush           unix  n       -       n       1000?   0      flush
proxymap        unix  -       -       n       -       -      proxymap
smtp            unix  -       -       n       -       -      smtp

# When relaying mail as backup MX, disable fallback_relay to avoid MX loops

relay       unix  -       -       n       -       -      smtp
          -o fallback_relay=
#       -o smtp_helo_timeout=5 -o smtp_connect_timeout=5

showq       unix  n       -       n       -       -      showq
error       unix  -       -       n       -       -      error
discard     unix  -       -       n       -       -      discard
local       unix  -       n       n       -       -      local
virtual     unix  -       n       n       -       -      virtual
lmtp        unix  -       -       n       -       -      lmtp
anvil       unix  -       -       n       -       1      anvil
scache      unix  -       -       n       -       1      scache

# maildrop. See the Postfix MAILDROP_README file for details.
# Also specify in main.cf: maildrop_destination_recipient_limit=1

maildrop    unix  -       n       n       -       -      pipe
  flags=DRhu user=vmail argv=/usr/local/bin/maildrop -d ${recipient}
```

```
cyrus      unix  -       n       n       -       -       pipe
  user=cyrus argv=/usr/lib/cyrus/bin/deliver -e ${sender} -m ${extension} ${user}

uucp       unix  -       n       n       -       -       pipe
  flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)

ifmail     unix  -       n       n       -       -       pipe
  flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)


bsmtp      unix  -       n       n       -       -       pipe
  flags=Fq. user=foo argv=/usr/local/sbin/bsmtp -f $sender $nexthop $recipient

procmail   unix  -       n       n       -       -       pipe
  flags=R user=nobody argv=/usr/bin/procmail -t -m /etc/procmailrc ${sender} ${recipient}

#amavis
smtp-amavis unix - - y - 2 smtp
localhost:10025 inet n - n - - smtpd
 -o content_filter=
 -o smtpd_recipient_restrictions=permit_mynetworks,reject
 -o mynetworks=127.0.0.0/8
```

Figura 23: /etc/postfix/master.cf

Cada entrada en el fichero es un conjunto de ocho campos separados por blancos o tabuladores, y cuyo orden y significado es:

Service	type	private	unprivileged	chroot	wakeup	maxprocess	command

Service: Nombre del servicio que se está configurando.

Type: Tipo de comunicación de transporte utilizado por el servicio.

Private: Restricciones de seguridad a procesos externos.

Unprivileged: Ejecución en modo no privilegiado.

Chroot: Indica si el servicio se ejecuta en un directorio de acceso restringido.

Wakeup: Segundos que deben transcurrir para que el proceso master despierte el servicio.

Maxprocess: Número máximo de procesos que puede usar el servicio.

Command: Nombre del programa a ejecutar y parámetros a pasar.

Service: Cada servicio de Postfix debe tener una entrada en el fichero master.cf, que son:

bounce: Devuelve al remitente los mensajes rechazados.

bsmtp: Encamina mensajes mediante el protocolo BSMTP.

cleanup: Procesa el correo entrante y soluciona posibles problemas en las direcciones de correo.

cyrus: Encamina correo mediante Cyrus Mail.

defer: Encamina mensajes fallidos o reintenta encaminar mensajes que estén en la cola defer.

error: Fuerza que un mail sea rechazado.

flush: Mantiene el control de los mensajes que están pendientes de encaminar.

ifmail: Encamina mensajes mediante ifmail.

lmtp: Encamina mensajes mediante el protocolo LMTP.

local: Encamina mensajes a usuarios locales.

pickup: Gestiona los mensajes que están esperando en la cola incoming.

qmgr: Procesa los mensajes que están en la cola incoming y decide cuál debe ser el método de encaminamiento.

relay: Recibe y encamina mensajes mediante el protocolo SMTP.

rewrite: Rescribe o verifica que las direcciones están en formato FQDN.

showq: Proporciona información sobre el estado de las colas.

smtp: Recibe y encamina mensajes mediante el protocolo SMTP.

uucp: Recibe y encamina mensajes mediante el protocolo UUCP.

Type:
Especifica el mecanismo utilizado por el proceso para comunicarse con otros módulos, que puede ser de tres tipos:

- ✧ Internet sockets (inet)
- ✧ Unix sockets (unix)
- ✧ Pipes con nombre (fifo)

Private:
Indica cuándo el canal de comunicación de un proceso debe estar accesible a procesos ajenos a Postfix.

Postfix utiliza dos subdirectorios: public y private, donde se crean los pipes con nombre de cada uno de los servicios, en función de que sean públicos o privados.

Unprivileged:
Especifica con qué privilegio de usuario se ejecuta el servicio. Si se especifica y (que es el valor por omisión) el servicio se ejecuta con los del usuario descrito en la directiva

mail_owner de main.cf, que por defecto es postfix. Si se indica n, el servicio se ejecuta con privilegios de root.

Chroot:
Se indica que el servicio se ejecuta en un entorno chroot, lo que proporciona niveles adicionales de seguridad. Esta opción se activa indicando y en este campo. Una única restricción: los procesos locales y pipe no pueden ejecutarse en modo chroot.

Wakeup:
Indica los segundos que deben transcurrir para que el proceso master envíe una señal para despertar el servicio correspondiente. En la actualidad sólo los servicios pickup, qmgr y flush utilizan esta directiva.

Existe una opción adicional, que es añadir el símbolo ? al final del valor que señala el intervalo. Con esto se indica al proceso master que sólo envíe la señal de despertar al servicio si este se está ejecutando. En la actualidad sólo flush soporta esta funcionalidad.

Maxprocess:
Especifica el número máximo de procesos que puede tener en ejecución el servicio. En caso de no indicarse nada se admiten 50 procesos.

Commands:
Determina el programa que debe ejecutarse para dar soporte al servicio definido.

Todos los programas admiten las siguientes opciones:

-v Habilita mayor detalle de log.
-D Activa el modo debug.

Usaremos el protocolo LMTP para la comunicación entre el MTA Postfix y Cyrus. Usaremos el socket /var/run/cyrus/socket/lmtp, por lo que debemos asegurarnos de que el servicio lmtpunix está habilitado en el fichero /etc/cyrus.conf y que Postfix tiene acceso a ese fichero (un socket, a efectos de permisos, funciona igual que un fichero).

Asimismo, Cyrus requiere que las entregas por LMTP estén autenticadas, y asume que las que se hagan a través del socket Unix son de confianza y las preautentica como si vinieran del usuario postman (ficticio).

Por lo tanto, nos aseguraremos de que el fichero /etc/postfix/master.cf contenga esta línea:

```
lmtp    unix  -       -       n       -       -       lmtp
```

En este archivo esta una seccion de los MDA con los que postfix puede comunicarse como son:

maildrop, cyrus, uucp, ifmail y procmail. ya que el MDA que postfix esta utillizando segun la configuracion es cyrus, hay que prestar especial atención a la ruta apuntada por el parámetro argv pues en alguna ocasión no refleja bien la situación del programa deliver y las entregas de correo pueden llegar a no efectuarse:

```
cyrus     unix  -       n       n       -       -       pipe
  user=cyrus argv=/usr/lib/cyrus/bin/deliver -e ${sender} -m ${extension} ${user}
```

Aquí agregamos el MDA llamado cyrus, además correrá con los privilegios de el usuario vmail.

NOTA: Deben de ir en dos líneas, la segunda inicia con espacio en blanco lo que indica que es continuación de la línea de arriba.

Al final del fichero están las líneas correspondientes al comunicación entre postfix y amavis-new estas líneas indican el puerto (10025) donde hemos habilitado smtpd, por el cual amavis-new va ha redirigir el mensaje ya una vez procesado.

```
queue_directory = /var/spool/postfix

command_directory = /usr/sbin

daemon_directory = /usr/lib/postfix

mail_owner = postfix

#default_privs = nobody

myhostname = mail.midominio.unanleon.edu.ni

mydomain = midominio.unanleon.edu.ni

myorigin = $mydomain

# RECEIVING MAIL
inet_interfaces = all

mydestination = midominio.unanleon.edu.ni

local_recipient_maps =
unknown_local_recipient_reject_code = 550

# The mailbox_transport specifies the optional transport in master.cf
#cyrus es el MDA
mailbox_transport = cyrus

# DEBUGGING CONTROL
debug_peer_level = 2

# The debugger_command specifies the external command that is executed
# when a Postfix daemon program is run with the -D option.
debugger_command =
        PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
        xxgdb $daemon_directory/$process_name $process_id & sleep 5

# INSTALL-TIME CONFIGURATION INFORMATION

sendmail_path = /usr/sbin/sendmail
newaliases_path = /usr/bin/newaliases
```

```
mailq_path = /usr/bin/mailq
setgid_group = maildrop
html_directory = /usr/share/doc/packages/postfix/html
manpage_directory = /usr/share/man
sample_directory = /usr/share/doc/packages/postfix/samples
readme_directory = /usr/share/doc/packages/postfix/README_FILES
inet_protocols = all
biff = no

#ALIAS Virtuales
virtual_alias_maps = hash:/etc/postfix/virtual, ldap:/etc/postfix/ldap-aliases.cf, hash:/etc/aliases,
hash:/var/lib/mailman/data/aliases

relocated_maps = hash:/etc/postfix/relocated
transport_maps = hash:/etc/postfix/transport
masquerade_exceptions = root
masquerade_classes = envelope_sender, header_sender, header_recipient
program_directory = /usr/lib/postfix

masquerade_domains =
defer_transports =
mynetworks_style = subnet
disable_dns_lookups = no
relayhost =
mailbox_command =

strict_8bitmime = no
disable_mime_output_conversion = no

#ALIAS
alias_maps = hash:/etc/aliases,ldap:/etc/postfix/ldap-aliases.cf
#cuota ilimitada
mailbox_size_limit = 0
message_size_limit = 10240000
smtpd_data_restrictions = reject_unauth_pipelining
best_mx_transport = local

#amavis
content_filter = smtp-amavis:127.0.0.1:10024
```

Figura 24: /etc/postfix/main.cf

Parámetros principales en nuestra configuración:

mail_owner: Propietario de las colas de mensajes.

myhostname: Representa el nombre del host es decir el nombre con el que aparece en el DNS el servidor de correo.

mydomain: Nombre del dominio principal, los dominios virtuales no se anotan en este parámetro.

myorigin: Nombre de el dominio con el que salen los correos.

inet_interfaces: Interfaces de red en la que escuchara peticiones.

mydestination: Dominios en los cuales recibe correo.

mailbox_size_limit = 0: Este parámetro significa que postfix permita una cuota ilimitada.

local_recipient_maps: Específica donde están almacenados los nombres de los usuarios.

unknown_local_recipient_reject_code = 450: Hace que si no encuentra en usuario local intenta más tarde y eso hace que se almacenen varios correos en cola.

mynetworks: A que redes se les permite hacer relay por este servidor.

relay_domains: A que dominios se les permite hacer relay.

alias_maps: Lugar donde están almacenados los alias de los correos.

mailbox_transport: Otro parámetro importante a tener en cuenta es el referente al *MDA*, o el programa que se encargará de transportar el correo de un programa a otro.

message_size_limit: Limite en tamaño de un mensaje (en bytes).

maximal_queue_lifetime: Tiempo que puede estar un mensaje en las colas.

content_filter = smtp-amavis:127.0.0.1:10024: Con esta línea haremos que *Postfix* redirija el tráfico hacia el puerto (10024) de *loopback* de *Amavisd-new*, que procesará el mensaje y lo redirigirá hacia el puerto 10025, donde hemos habilitado smtpd.

```
postmaster: postmaster
configdirectory: /var/lib/imap
partition-default: /var/spool/imap
sievedir: /var/lib/sieve
admins: cyrus
allowanonymouslogin: no
allowplaintext: 1
sasl_mech_list: PLAIN
autocreatequota: 20480
reject8bit: no
quotawarn: 90
timeout: 30
poptimeout: 10
dracinterval: 0
drachost: localhost
sasl_pwcheck_method: saslauthd
lmtp_overquota_perm_failure: no
lmtp_downcase_rcpt: yes
```

Figura 25: /etc/imapd.conf

/etc/imapd.conf Es el fichero de configuración del servidor Cyrus IMAP y en él se definen los parámetros locales para IMAP. Cada una de las líneas de tiene el formato *opción:*

valor, donde *opción* es el nombre de la opción a configurar y *valor* el valor al cual se está estableciendo esa opción. A continuación se detallan algunas de las opciones más relevantes y sus valores recomendados:

lmtp_downcase_rcpt: Esta opción, que sirve para forzar que el nombre de usuario se convierta a minúsculas, viene por defecto comentada, es decir, con valor *no*. Debido a que Cyrus diferencia mayúsculas y minúsculas, es una buena idea trabajar con los nombres de usuario siempre en minúsculas (el valor por defecto asume que el usuario es consciente de lo que está haciendo).

configdirectory, partition-default: Son respectivamente los directorios donde se encuentran las configuraciones de los buzones, y donde se encuentran buzones, las particiones de los busones.

admins: Esta opción permite definir los usuarios que tendrán permisos de administrador (flag *a* de la ACL de un buzón) sobre todos los buzones del sistema. El usuario *cyrus*, y únicamente él, es la opción más recomendable, por lo que bastará con descomentar la línea del fichero. Este usuario se va a autenticar mediante el método SASL, por lo que debe añadirse a la base de datos */etc/sasldb2* mediante el comando *saslpasswd2*, tal que $saslpasswd2 -c cyrus.

allowanonymouslogin: Esta opción permite el acceso anónimo a los buzones en cuyas ACLs se haya añadido al usuario *anonymous*. Carece de sentido a menos que se quieran implementar grupos de noticias, por lo que se dejará su valor por defecto *no*.

sasl_pwcheck_method: El metodo de autentificacion que usaremos para comprobar los passwords, por ejemplo auxprop (la base de datos de cyrus por defecto), saslauthd (contra el demonio saslauthd), etc. Nosotros, para autenticar contra ldap usaremos saslauthd.

allowplaintext: Mediante esta opción decidimos si vamos a permitir uso del mecanismo de autentificación *sasl plain*. Es recomendable mantener el valor por defecto.

sasl_mech_list: Esta es la lista de los mecanismos de autenticación que se van a soportar. Es útil para evitar que se prueben todos los plugings existentes y para definir el orden de los mismos. En nuestro caso utilizamos PLAIN.

autocreatequota: Especificación de la cuota en general medida en kb, pero se puede cambiar por ejemplo con la herramienta cyradm usando el comando sq ejemplo:

```
cyradm --u cyrus --s localhost --auth plain
localhost> sq user.arlen 25000
```

quotawarn: El por ciento de utilización de cuota sobre los que el servidor genera advertencias.

poptimeout: el tiempo en minutos que puede estar el servidor Pop inactivo, es decir si se agota este tiempo y el servidor Pop no recibe ninguna petición aborta la conexión.

timeout: el tiempo en minutos que puede estar el servidor Imap inactivo, si no le llega ninguna petición durante este tiempo aborta la conexión.

LDAP.

```
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         /usr/share/open-xchange/openxchange.schema
include         /etc/openldap/schema/misc.schema
#include        /etc/openldap/schema/mail.schema

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

loglevel 256

modulepath      /usr/lib/openldap/modules

#sample acces control global
include /etc/openldap/acl_ox.conf

########################################################################
# BDB database definitions
########################################################################

database        bdb
suffix          "dc=midominio,dc=unanleon,dc=edu,dc=ni"
checkpoint      1024 5
#cachesize       10000
rootdn          "uid=mailadmin,dc=midominio,dc=unanleon,dc=edu,dc=ni"
rootpw          "secret"

directory       /var/lib/ldap
index   objectClass     eq

replogfile /var/lib/ldap/trams.log

# should be about the number of entries in the database
cachesize       80000
# 700 MB memory cache size for database info
dbcachesize     734003200
################
### INDEXES ###
################
index uid,mailEnabled,cn,sn,givenname,lnetMailAccess,alias,loginDestination eq,sub

replogfile      /var/lib/ldap/replog
```

Figura 26: /etc/openldap/slapd.conf

Directivas globales:

La primera directiva de slapd.conf, que se muestra en el Ejemplo “slapd.conf: directiva de inclusión para esquemas”, especifica el esquema con el que se organiza el directorio LDAP. La entrada core.schema es obligatoria. Al final de esta directiva se añaden esquemas necesarios adicionales.

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

Estos dos archivos contienen el PID (ID de proceso) y algunos de los argumentos con los que ha comenzado el proceso slapd. No es necesario realizar ninguna modificación aquí.

Control de acceso global (Acl):

```
include /etc/openldap/acl_ox.conf
```

En el archivo acl_ox.conf se regula los permisos de acceso para el directorio LDAP en el servidor. Los ajustes que se realicen aquí en la sección global de slapd.conf son válidos mientras que no se declaren reglas de acceso personalizadas en la sección específica de la base de datos. Dichas reglas sobrescribirían las declaraciones globales. La regulación del control de acceso en LDAP es un proceso extremadamente complejo. Las siguientes sugerencias pueden resultar útiles:

• Cada regla de acceso cuenta con la siguiente estructura:
  access to <qué> by <quién> <acceso>

qué: Es un espacio reservado para el objeto o atributo al que se otorga acceso.

Las ramas individuales del directorio se pueden proteger explícitamente con reglas independientes. También es posible procesar zonas del árbol de directorios con una regla mediante el uso de expresiones regulares. slapd evalúa todas las reglas en el orden en el que se muestran en el archivo de configuración. Las reglas más generales se deberían mostrar después de las más concretas (la primera regla que slapd considera válida se evalúa y las entradas siguientes se omiten.

quién: Determina a quién se debería otorgar acceso a las áreas determinadas con qué. Se pueden usar expresiones regulares. slapd aborta la evaluación de quién después de la primera coincidencia, de modo que las reglas más específicas deben encontrarse antes de las más generales.

Ejemplos del parámetro quien:
* Todos los usuarios sin excepción

anonymous: Usuarios no autenticados (“anónimos”)
users: Usuarios autenticados
self: Usuarios conectados con el objeto de destino
dn.regex=<regex>: Todos los usuarios que coinciden con la expresión regular.

access: Especifica el tipo de acceso. Utilice las opciones que se muestran a continuación:

none: Sin acceso.
auth: Para contactar con el servidores.
compare: A objetos para acceso de comparación.
search: Para el empleo de filtros de búsqueda.
read: Acceso de lectura.
write: Acceso de escritura.

Directivas específicas de base de datos en slapd.conf:

database bdb: El tipo de base de datos, una de Berkeley en este caso.

checkpoint: Determina la cantidad de datos (en kb) que se mantiene en el registro de transacciones antes de que se escriban en la base de datos real y el tiempo (en minutos) entre dos acciones de escritura.

cachesize: Define el número de objetos mantenidos en el caché de la base de datos.

suffix: Determina la parte del árbol de LDAP de la que el servidor es responsable.

rootdn: Aquí se determina quién posee los derechos de administrador de este servidor. El usuario que aparezca aquí no necesita tener una entrada LDAP ni existir como usuario normal.

rootpw: La contraseña del administrador está definida con rootpw. En lugar de usar aquí secret, es posible introducir el algoritmo hash de la contraseña del administrador creado por slappasswd.

directory: La directiva directory indica el directorio (en el sistema de archivos) en el que los directorios de la base de datos se almacenan en el servidor.

index: La última directiva, index objectClass eq, da como resultado el mantenimiento de un índice en todas las clases de objetos. Los atributos que los usuarios buscan con más frecuencia pueden añadirse aquí según su uso.

Nota: Las reglas personalizadas de acceso definidas para la base de datos se usan en lugar de las reglas globales de acceso.

```
# LDAP Defaults
#

# See ldap.conf(5) for details
# This file should be world readable but not world writable.

BASE    dc=midominio,dc=unanleon,dc=edu,dc=ni
URI     ldap://ldapserver.midominio.unanleon.edu.ni

#SIZELIMIT      12
#TIMELIMIT      15
#DEREF          never
TLS_REQCERT allow
```

Figura 27: /etc/openldap/ldap.conf

BASE: Este parámetro representa el nodo raíz de árbol ldap, hay que tener cuidado de no dejar espacios en blancos después de la coma ya que puede llegar a ocasionar errores.

URI: Aquí se especifica la conexión al servidor ldap.

PostgreSQL.

Primeramente editar este archivo /etc/sysconfig/postgresql y buscar un parámetro llamado /POSTGRES_OPTIONS e igualarlo a la opción "-i" la cual significa que para acceder a postgresql lo haremos con conexiones vía TCP/IP.

El comportamiento de PostgreSQL en nuestro sistema se puede controlar con tres ficheros de configuración. Estos tres ficheros son:

- **pg_hba.conf**: Este fichero se utiliza para definir los diferentes tipos de accesos que un usuario tiene en el cluster.

- **pg_ident.conf**: Este fichero se utiliza para definir la información necesaria en el caso que utilicemos un acceso del tipo ident en pg_hba.conf.

postgresql.conf: En este fichero podemos cambiar todos los parámetros de configuracion que afectan al funcionamiento y al comportamiento de PostgreSQL en nuestra maquina.

```
# "local" is for Unix domain socket connections only
local   all         all                               trust
# IPv4 local connections:
host    all         all         127.0.0.1/32          trust
# IPv6 local connections:
host    all         all         ::1/128               trust
```

Figura 28: /var/lib/pgsql/data/pg_hba.conf

Este fichero se utiliza para definir como, donde y desde que sitio un usuario puede utilizar nuestro servidor PostgreSQL. Todas las líneas que empiecen con el carácter # se interpretan como comentarios. El resto debe de tener el siguiente formato:

[Tipo de conexion] [database] [usuario] [IP] [Netmask] [Tipo de autentificacion] [opciones]

Dependiendo del tipo de conexión y del tipo de autentificación, [IP],[Netmask] y [opciones] pueden ser opcionales. El tipo de conexión puede tener los siguientes valores, local, host, hostssl y hostnossl. El tipo de método de autenticación puede tener los siguientes valores, trust, reject, md5, crypt, password, krb5, ident, pam o ldap.

El parámetro DATABASE: puede tomar el valor "all", "sameuser", "samerole", o el nombre de una base de datos cualesquiera.

El parámetro USER puede tomar los siguientes valores: "all" o el nombre de un usuario.

```
# - Connection Settings -

listen_addresses = 'localhost'          # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost', '*' = all
port = 5432
```

```
max_connections = 100

# - Memory -

shared_buffers = 1000                   # min 16 or max_connections*2, 8KB each

# - Where to Log -

log_destination = 'stderr'              # Valid values are combinations of
                                        # stderr, syslog and eventlog,
                                        # depending on platform.

# This is used when logging to stderr:
redirect_stderr = on                    # Enable capturing of stderr into log
                                        # files

log_rotation_age = 0                    # Automatic rotation of logfiles will
                                        # happen after so many minutes.  0 to
                                        # disable.
log_rotation_size = 10240               # Automatic rotation of logfiles will
                                        # happen after so many kilobytes of log
                                        # output.  0 to disable.

silent_mode = on                        # DO NOT USE without syslog or
                                        # redirect_stderr

# - What to Log -

#debug_print_parse = off
#debug_print_rewritten = off
#debug_print_plan = off
#debug_pretty_print = off
#log_connections = off
#log_disconnections = off
#log_duration = off
log_line_prefix = '%t %d %u '           # Special values:
                                        #   %u = user name
                                        #   %d = database name
                                        #   %r = remote host and port
                                        #   %h = remote host
                                        #   %p = PID
                                        #   %t = timestamp (no milliseconds)
                                        #   %m = timestamp with milliseconds
                                        #   %i = command tag
                                        #   %c = session id
                                        #   %l = session line number
                                        #   %s = session start timestamp
                                        #   %x = transaction id
                                        #   %q = stop here in non-session
                                        #        processes
                                        #   %% = '%'
                                        # e.g. '<%u%%%d> '

#---------------------------------------------------------------------------
# AUTOVACUUM PARAMETERS
#---------------------------------------------------------------------------
```

```
autovacuum = on                                 # enable autovacuum subprocess?

#---------------------------------------------------------------------------
# CLIENT CONNECTION DEFAULTS
#---------------------------------------------------------------------------

# These settings are initialized by initdb -- they might be changed
lc_messages = 'es_ES.UTF-8'                     # locale for system error message
lc_monetary = 'es_ES.UTF-8'                     # locale for monetary formatting
lc_numeric = 'es_ES.UTF-8'                      # locale for number formatting
lc_time = 'es_ES.UTF-8'                         # locale for time formatting
```

max_connections: Número máximo de clientes conectados a la vez a nuestras bases de datos. Deberíamos de incrementar este valor en proporción al número de clientes concurrentes en nuestro cluster PostgreSQL. Un buen valor para empezar es el 100.

shared_buffers: Este parámetro es importantísimo y define el tamaño del buffer de memoria utilizado por PostgreSQL. No por aumentar este valor mucho tendremos mejor respuesta. En un servidor dedicado podemos empezar con un 25% del total de nuestra memoria. Nunca mas de 1/3 (33%) del total. Por ejemplo, en un servidor con 4Gbytes de memoria, podemos usar 1024MB como valor inicial.

work_mem: Usada en operaciones que contengan ORDER BY, DISTINCT, joins. En un servidor dedicado podemos usar un 2-4% del total de nuestra memoria si tenemos solamente unas pocas sesiones (clientes) grandes. Como valor inicial podemos usar 8 Mbytes.

maintenance_work_mem: Usada en operaciones del tipo VACUUM, ANALYZE, CREATE INDEX, ALTER TABLE, ADD FOREIGN KEY. Su valor dependerá mucho del tamaño de nuestras bases de datos. Por ejemplo, en un servidor con 4Gbytes de memoria, podemos usar 256MB como valor inicial.

effective_cache_size: Parámetro usado por el 'query planner' de nuestro motor de bases de datos para optimizar la lectura de datos. En un servidor dedicado podemos empezar con un 50% del total de nuestra memoria. Como máximo unos 2/3 (66%) del total. Por ejemplo, en un servidor con 4Gbytes de memoria, podemos usar 2048MB como valor inicial.

checkpoint_segments: Este parámetro es muy importante en bases de datos con numerosas operaciones de escritura (insert,update,delete). Para empezar podemos empezar con un valor de 64. En grandes databases con muchos Gbytes de datos escritos podemos aumentar este valor hasta 128-256.

Apache.

En versiones anteriores la mayoría de la configuración de apache se encontraba en el archivo:

/etc/apache2/httpd.conf, pero en esta versión la configuración esta dividida en varios archivos que iremos explicando poco a poco. La configuración aquí comentada es bien básica:

/etc/apache2/httpd.conf

Buscar el parámetro DirectoryIndex y fijarse que tenga los siguientes valores:

DirectoryIndex index.html index.php: Lo cual significa que es capas de servir páginas html y también tiene soporte para php.

Dominios virtuales: Este es el caso de usar dominios virtuales. Es decir que una maquina se conocida con diferentes nombres pero que estos nombres estén asociados al mismo numero ip. Esto con el objetivo de que cuando acceda a este host este presente diferentes paginas Web dependiendo del nombre que se halla usado.

Para configurar dominios virtuales tiene que hacer lo siguiente:

Edite el archivo /etc/apache2/listen.conf y descomente el parámetro

NameVirtualHost de tal forma que quede de la siguiente manera:

NameVirtualHost *:80, donde el * es un comodín que puede ser reemplazado por una dirección ip.

En el directorio /etc/apache2/vhosts.d/ esta dos archivos que nos sirven de plantilla para realizar nuestros dominios virtuales: vhost-ssl.template vhost.template, donde la primer plantilla es para las conexiones vía https y la segunda es para las conexiones vía http.

Los principales parámetros son:

ServerAdmin: Este termino define la dirección de correo de la persona que se va ha encargar de darle mantenimiento al dominio virtual.

ServerName: Este define el nombre para este dominio virtual.

DocumentRoot: Aquí se definirá el lugar dentro del sistema de archivos donde residiran las paginas para este dominio virtual.

La directiva *Alias* permite redireccionar a un directorio que puede estar fuera del árbol de directorios especificado en *DocumentRoot*.

Con la directiva *Directory* definimos opciones que se aplican al directorio indicado y sus subdirectorios. Lo habitual es configurar unos permisos por defecto muy restrictivos:

- ✓ La directiva *AllowOverride* controla qué opciones pueden sobrescribirse con un

archivo *.htaccess .* Puede impedirse la modificación con *None* o permitirse con *All*.

- ✓ Con las directivas *Order* y *Allow* se controla el acceso al servidor utilizando el nombre de dominio o la IP del cliente. La directiva *Order allow, deny* determina que primero se evalúa la lista *Allow* y luego la *Deny*. La directiva *Allow from all* permite acceder a todo el mundo.

Es necesario configurar Apache para que se comunique con Tomcat. Para ello usaremos mod_jk.

Editamos **/etc/sysconfig/apache2** y agregamos **jk** a la lista de módulos APACHE_MODULES="...jk".

Imágenes de SquirrelMail.

Figura 29: Pantalla de inicio.

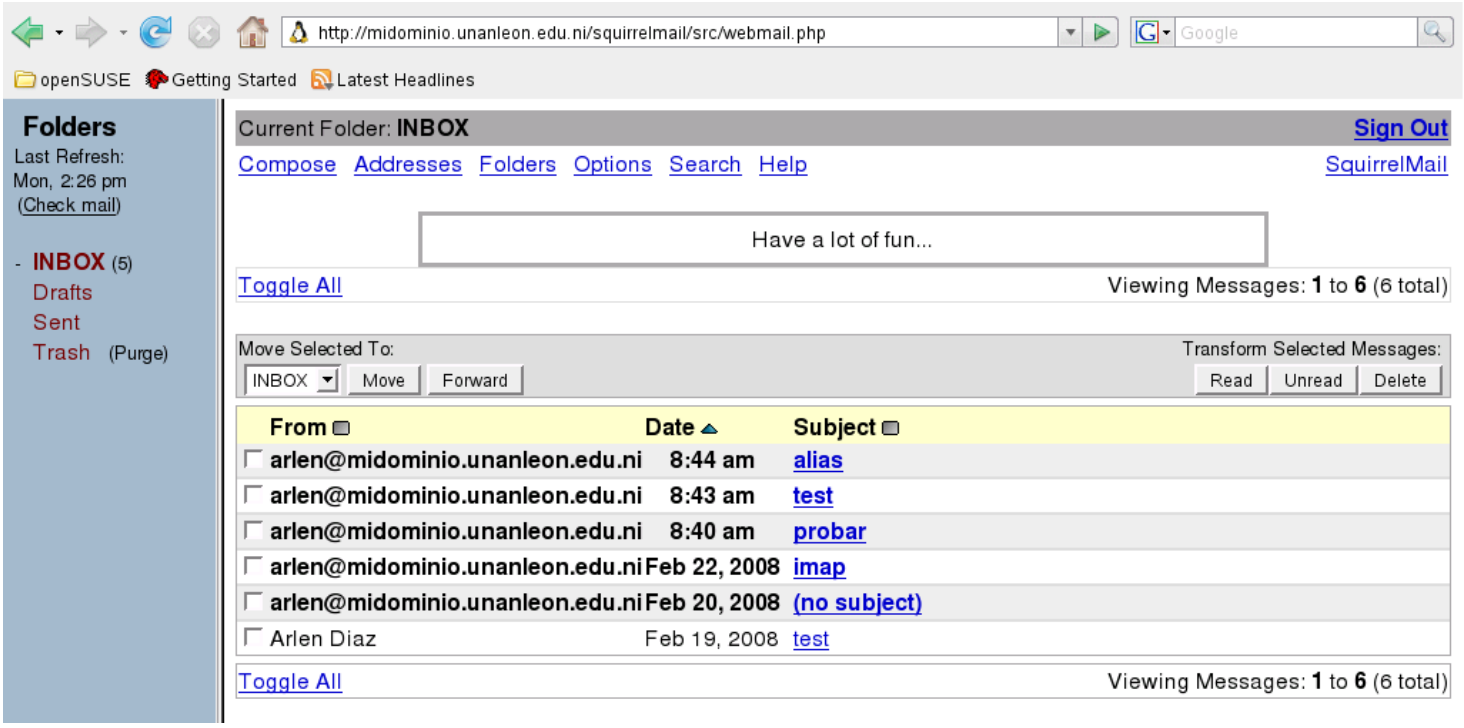

Figura 30: Buzón de mensajes recibidos.

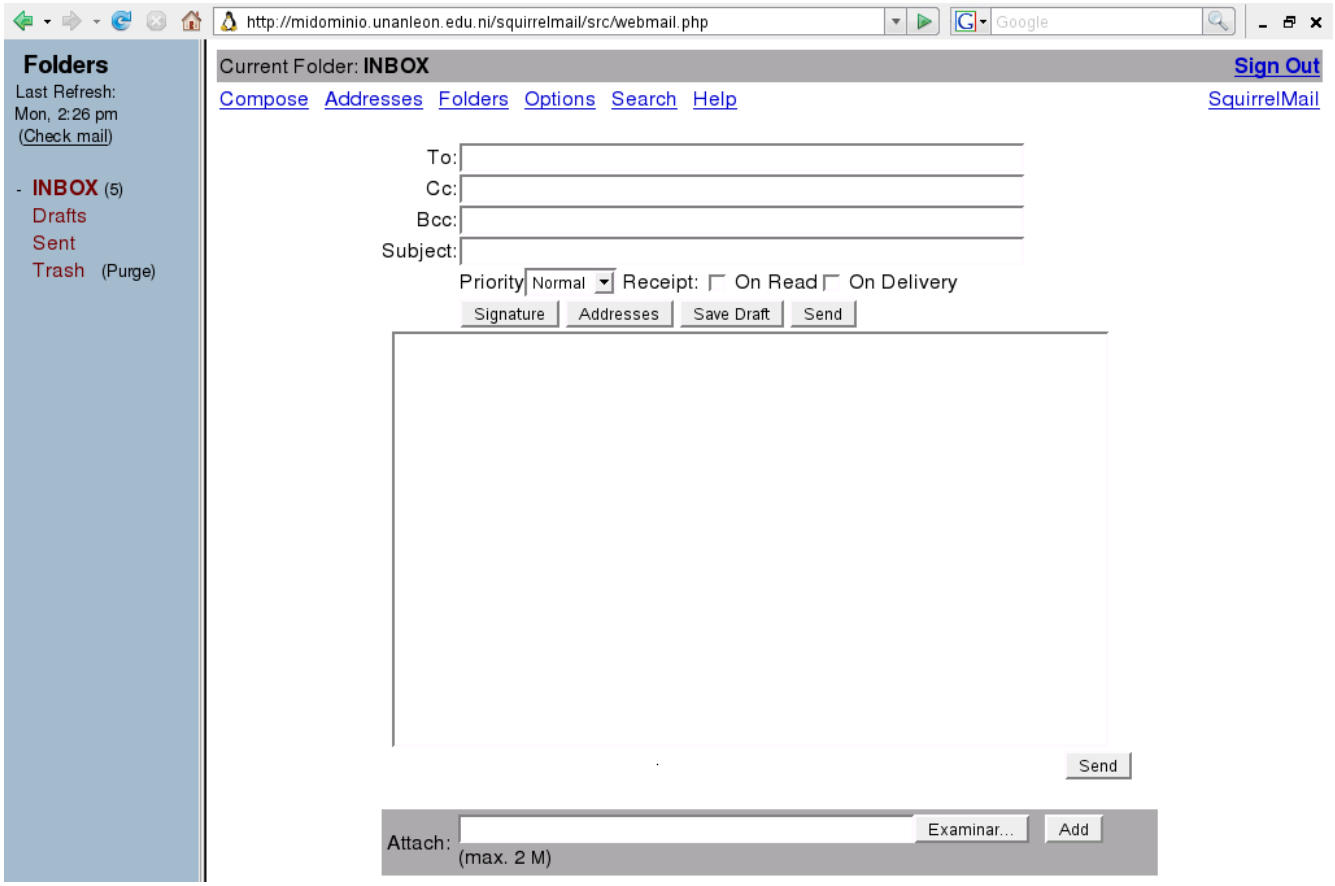

Figura 31: Formulario para enviar mensajes de correo.

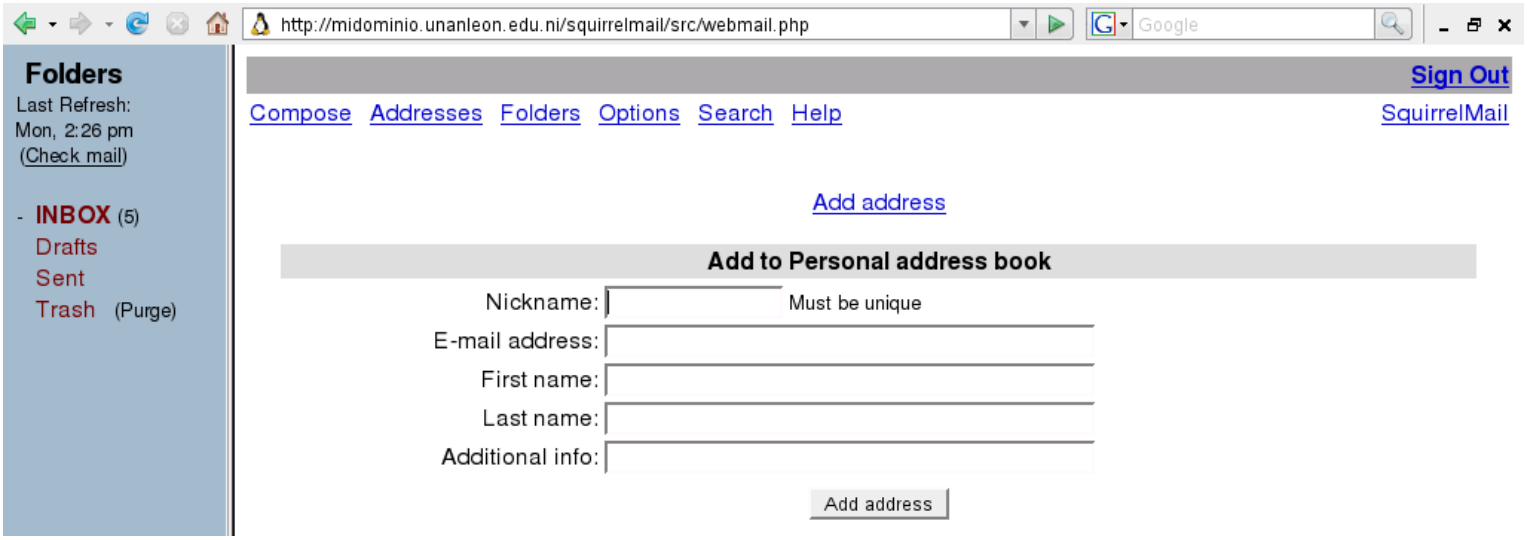

Figura 32: Formulario para agregar un nuevo contacto.

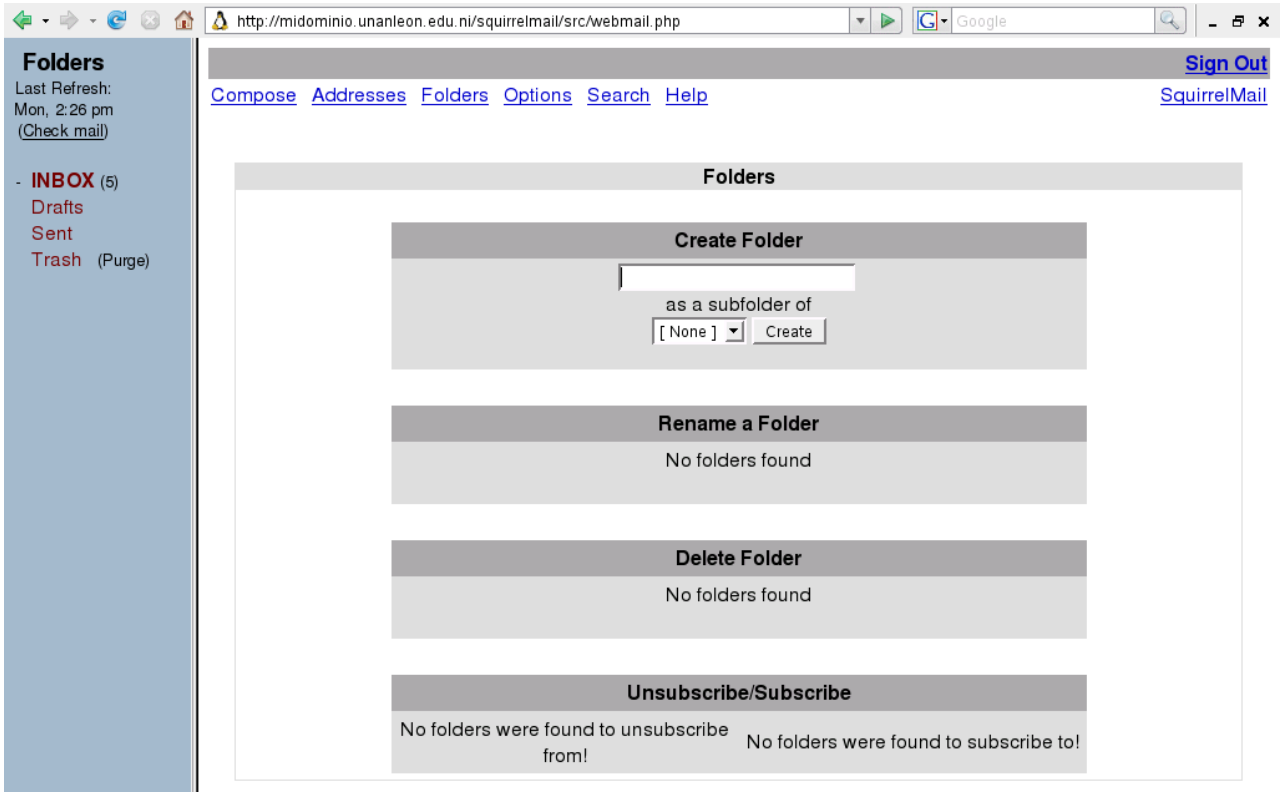

Figura 33: Formulario para crear directorios.

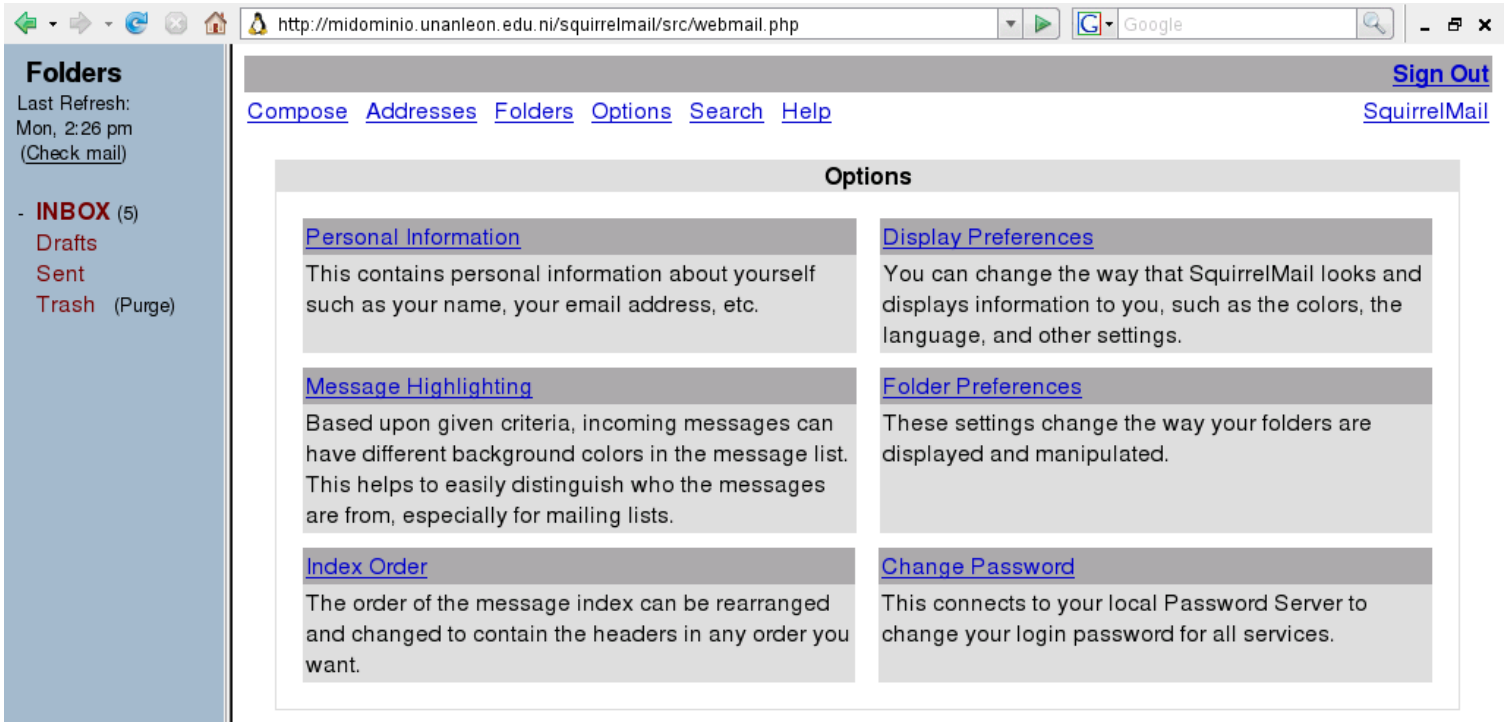

Figura 34: Formulario que muestra otros componentes de SquirrelMail.

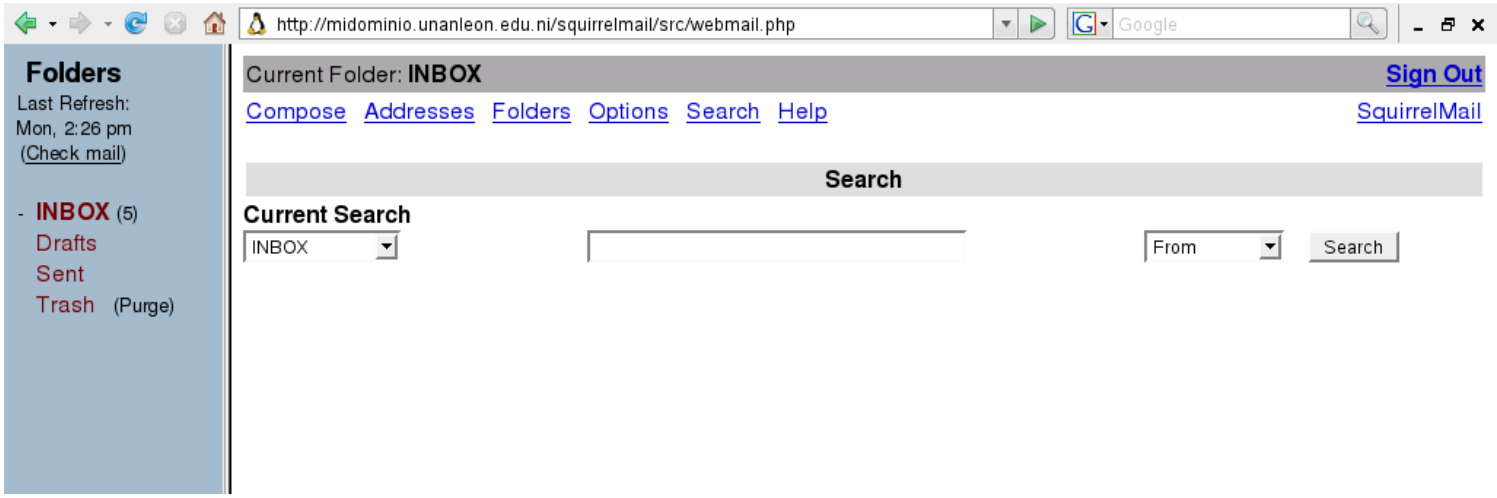

Figura 35: Formulario que busca directorios y archivos.

Imágenes de Open-Xchange.

Figura 36: Pantalla de Inicio.

Modulo de Open-Xchange.

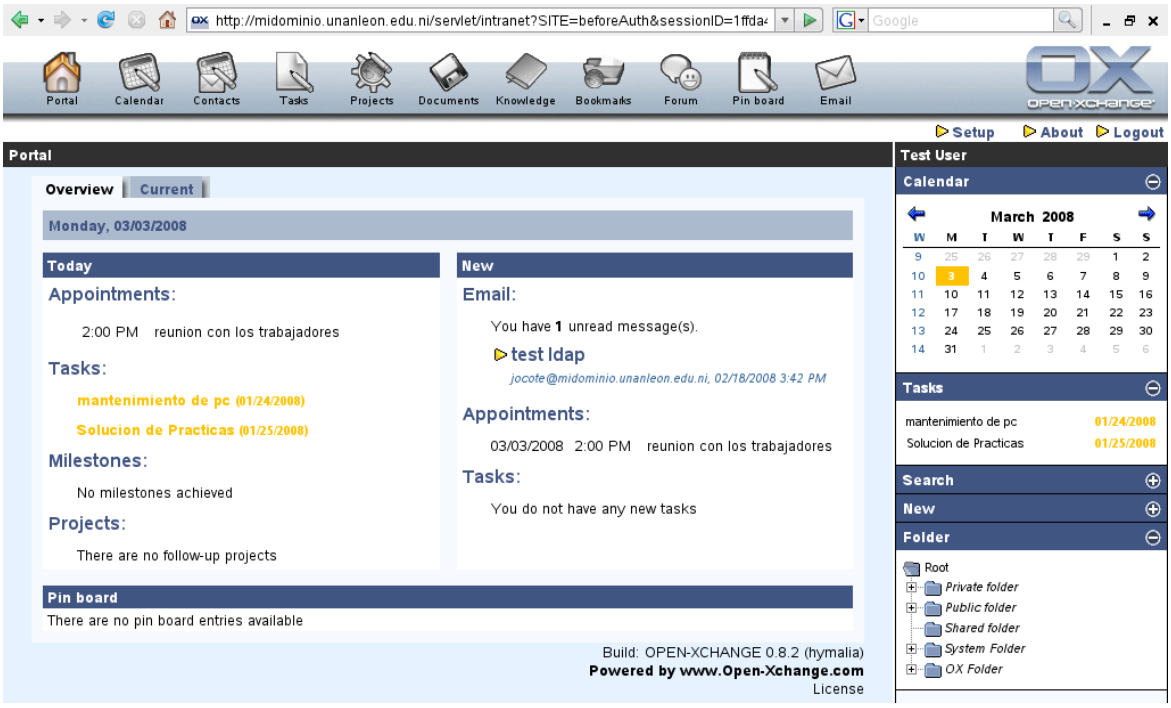

Figura 37: Portal.

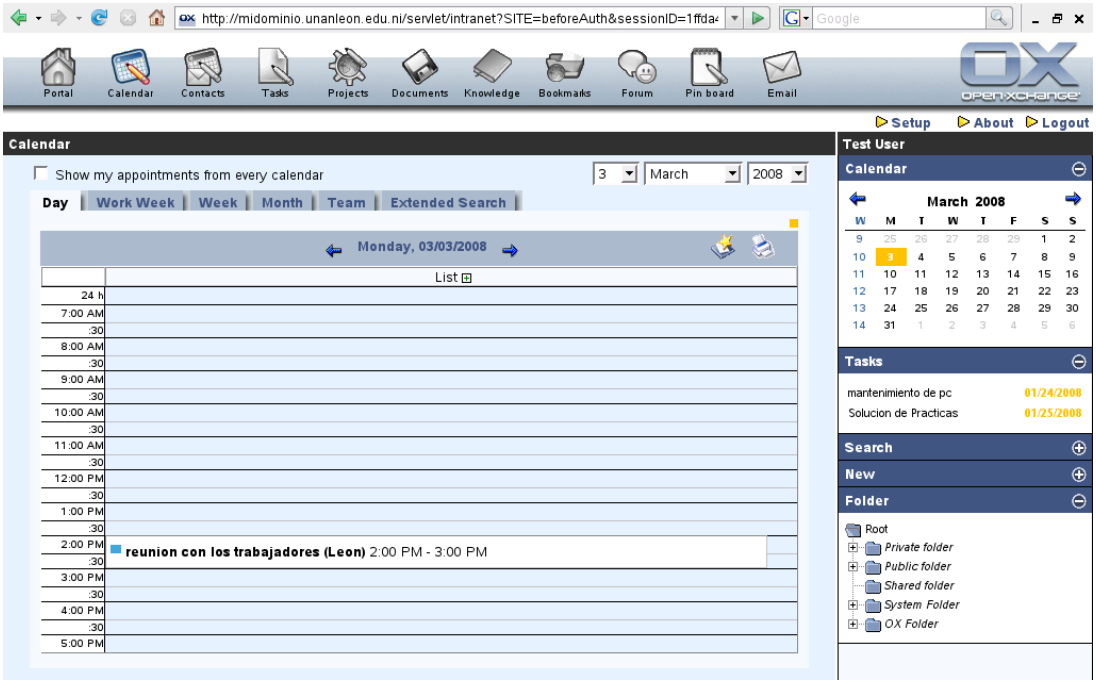

Figura 38: Calendario.

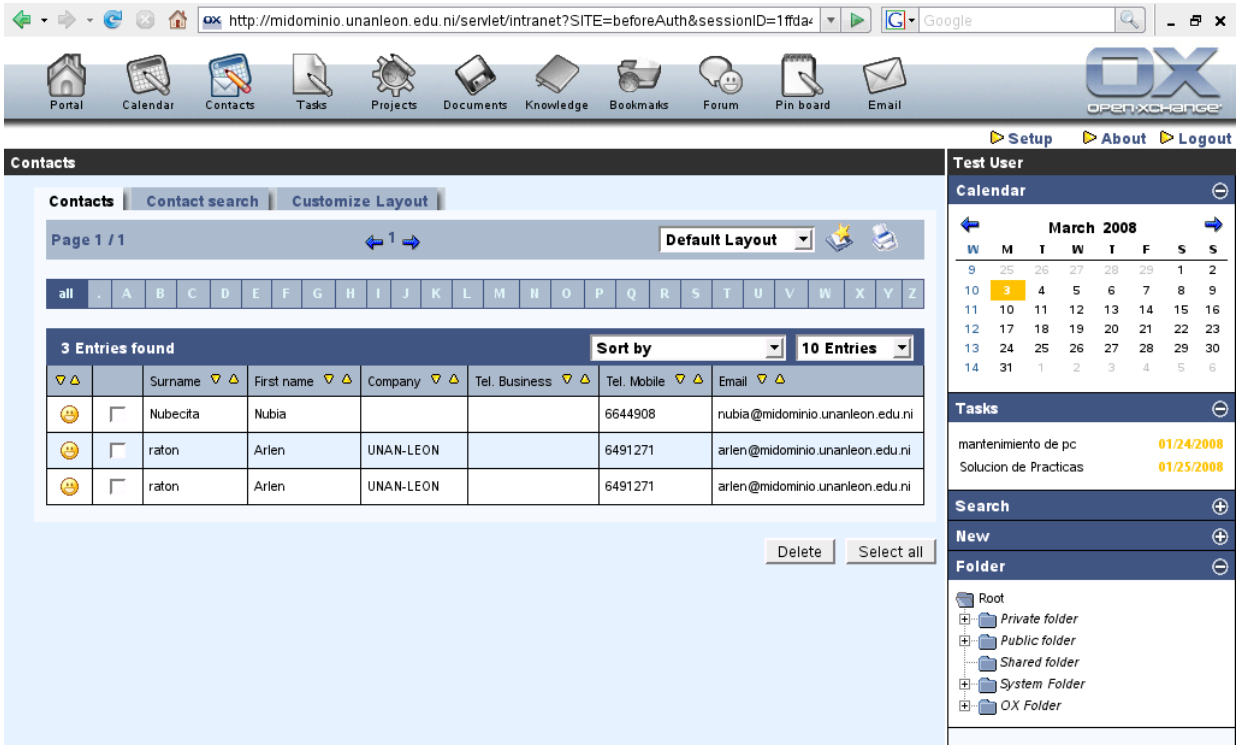

Figura 39: Contactos.

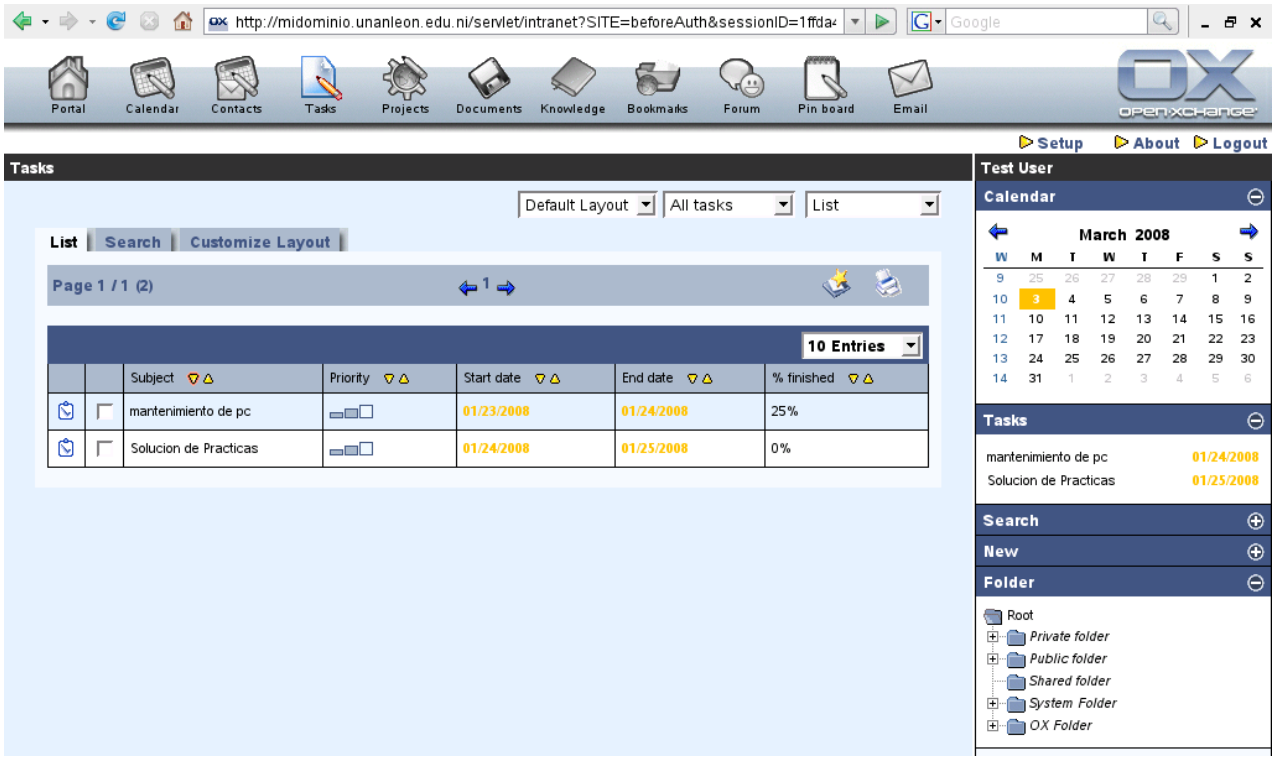

Figura 40: Tarea.

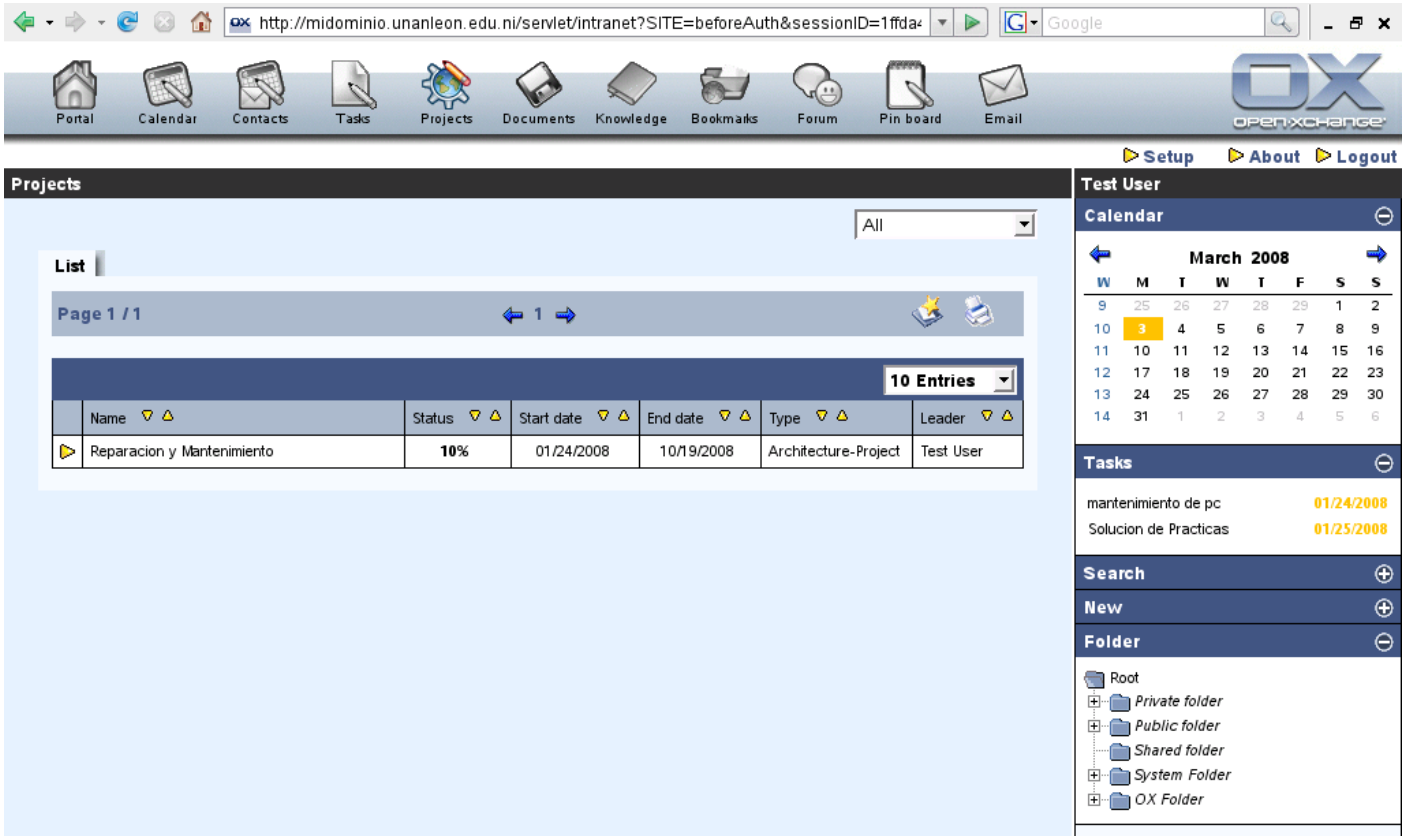

Figura 41: Proyectos.

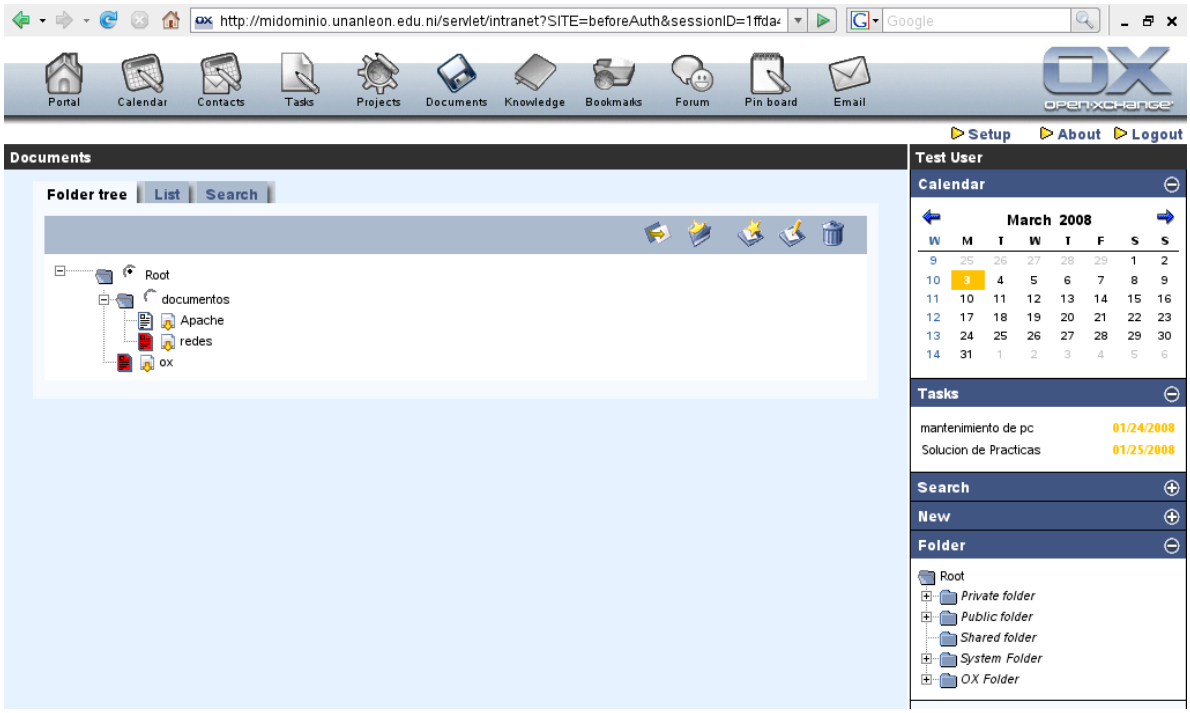

Figura 42: Documentos.

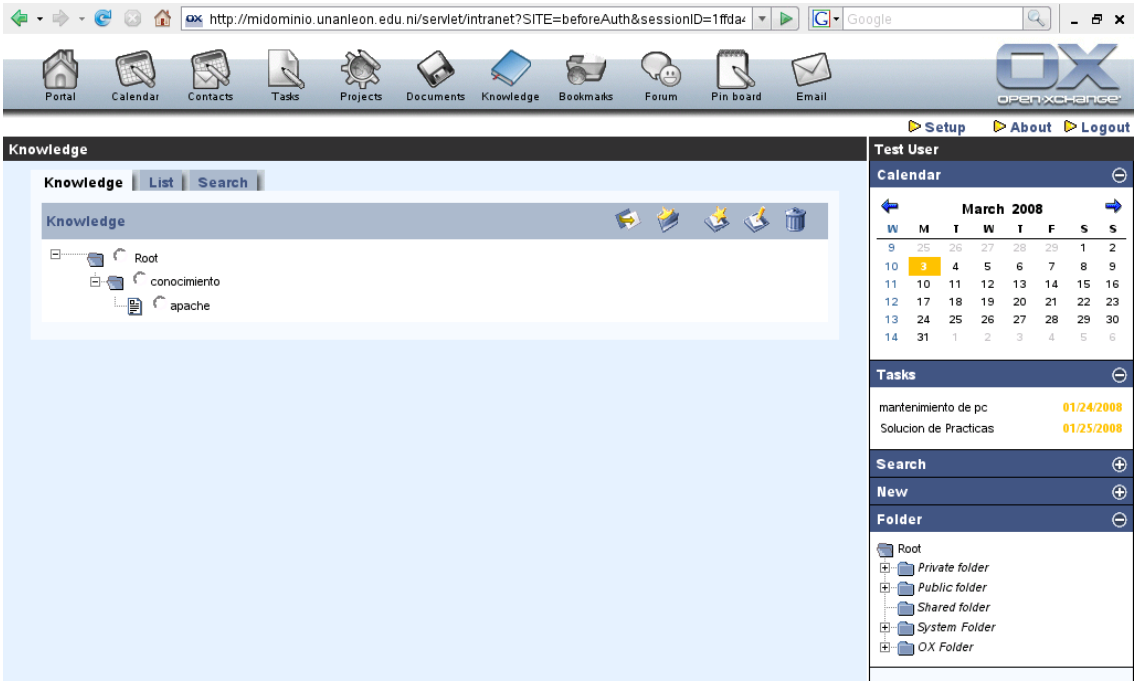

Figura 43: Conocimientos.

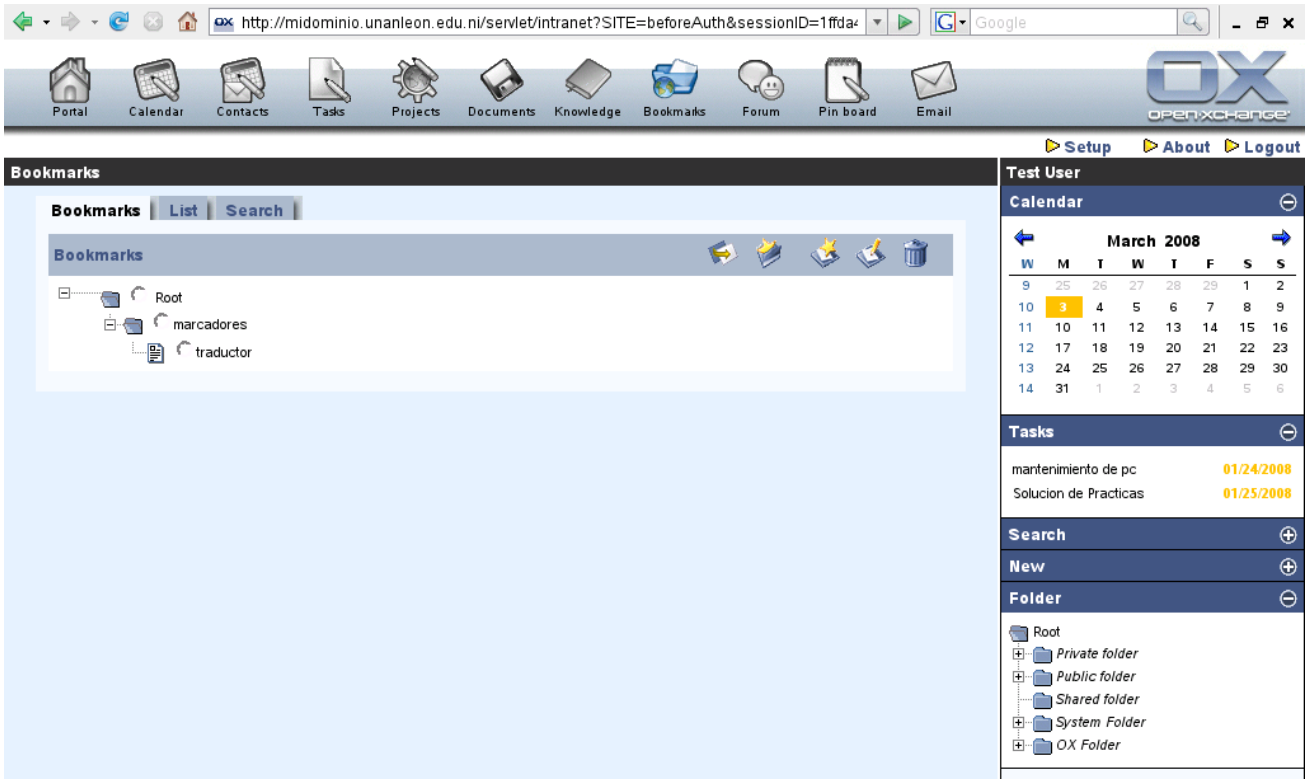

Figura 44: Favoritos.

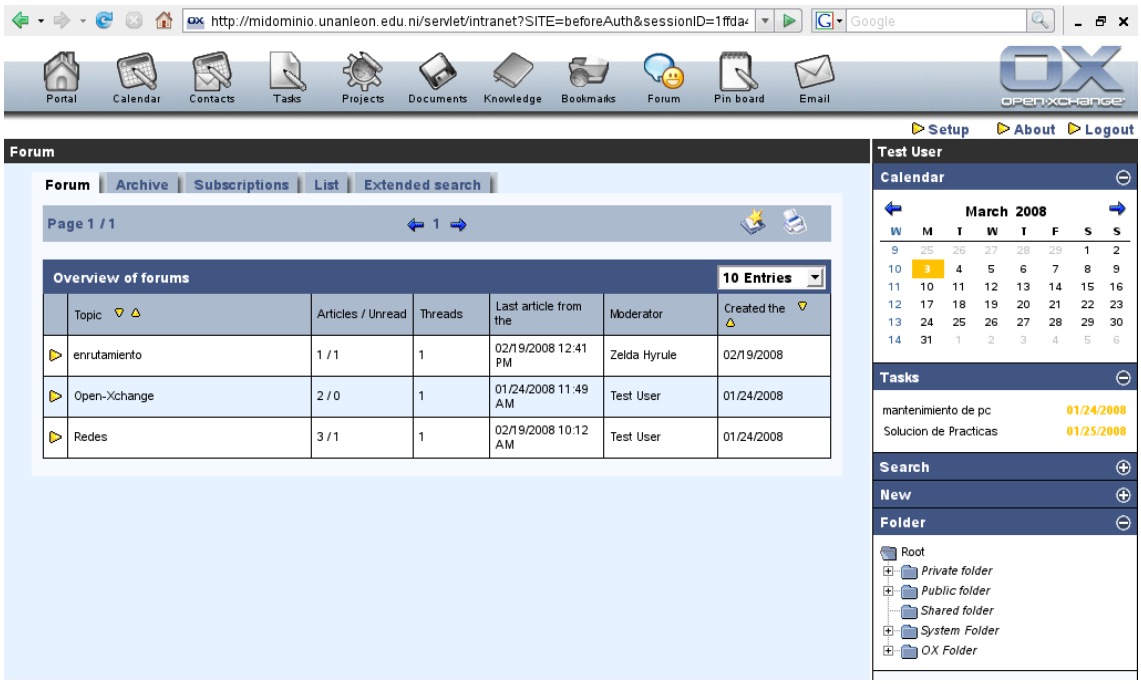

Figura 45: Foros.

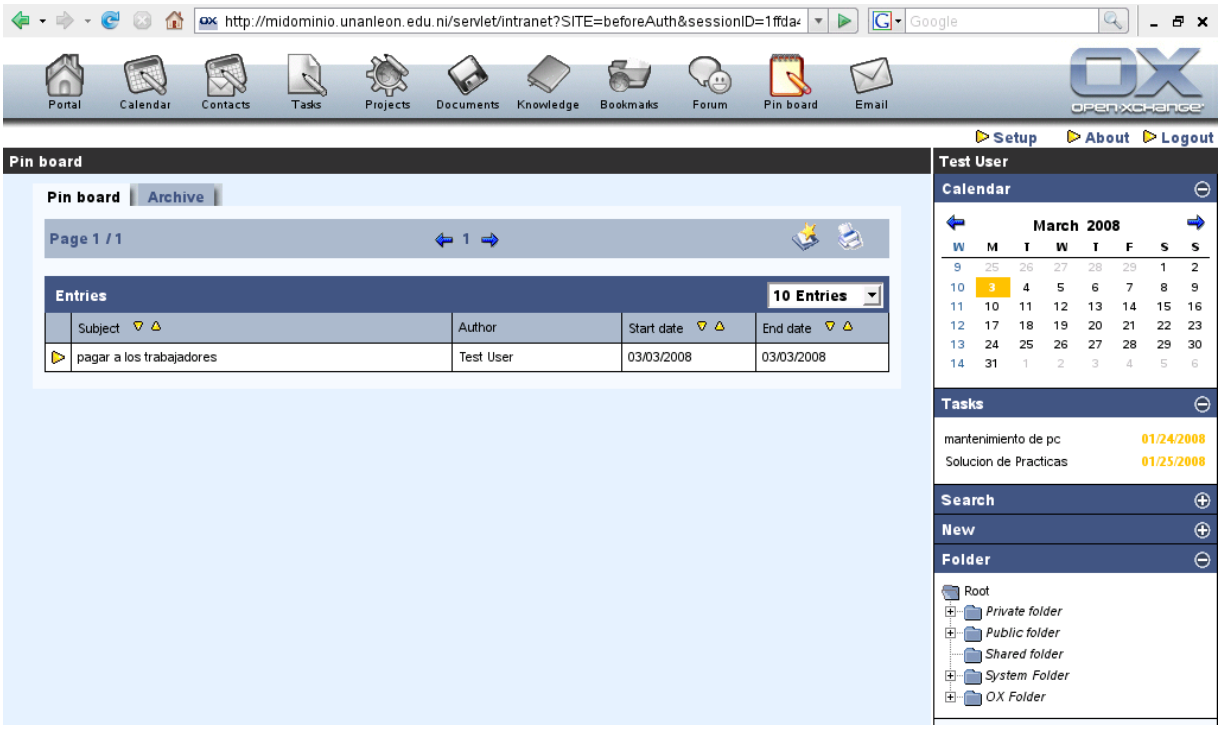

Figura 46: Tablero.

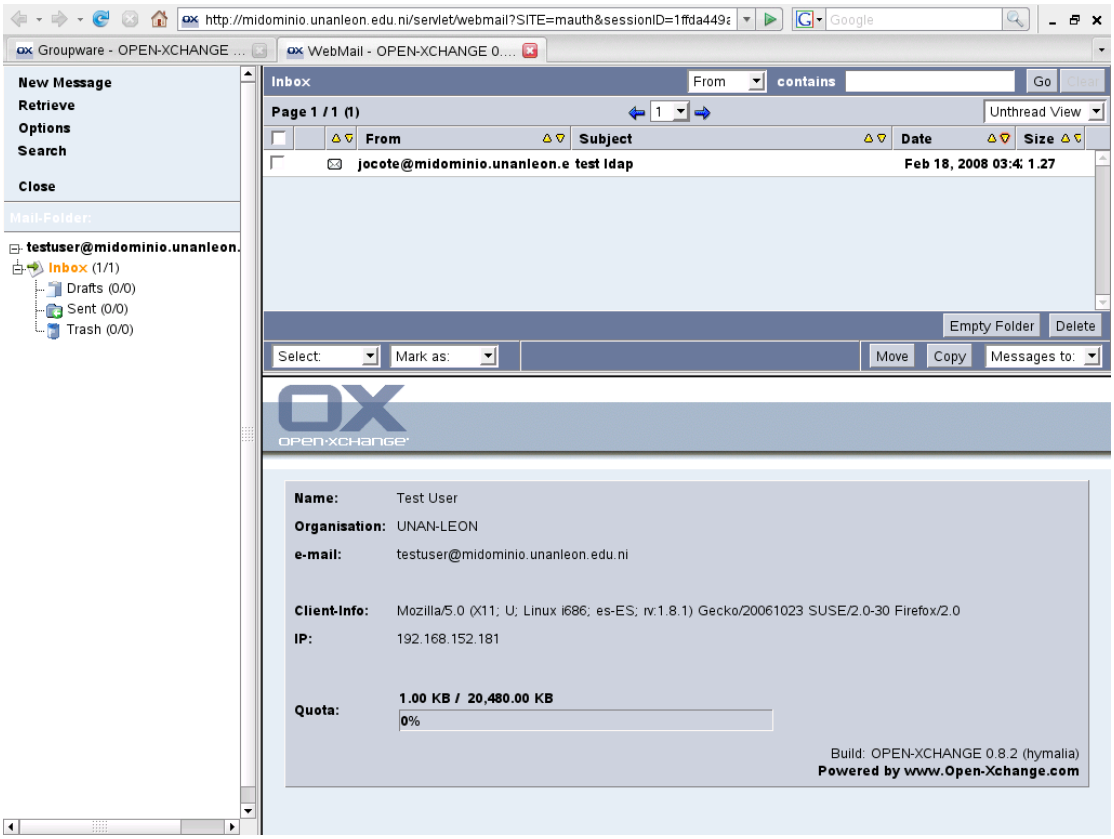

Figura 47: Correo.

XII. GLOSARIO.

- ✓ **API:** Interfase de Programación de la Aplicación. Especificación de las convenciones para llamar funciones, que define una interfase hacia un servicio.

- ✓ **Bifurcación:** es una característica que permite a una colección de archivos desarrollarse en dos o más rutas de acceso divergentes.

- ✓ **BIND:** Berkeley Internet Name Domain, anteriormente: Berkeley Internet Name Daemon, es el servidor de DNS más comúnmente usado en Internet, especialmente en sistemas Unix, en los cuales es un standard de facto.

- ✓ **BSD:** Distribución Estándar de Berkeley. Termino que describe cualquiera de los sistemas operativos tipo Unix basados en el sistema operativo UC Berkeley BSD.

- ✓ **Correo:** Electrónico (e-mail): Aplicación de red muy popular, en la que se trasmiten electrónicamente mensajes de correo entre usuarios terminales a través de varios tipos de redes, mediante varios protocolos de red. Se suele llamar e-mail.

- ✓ **Dirección IP:** Es una dirección de 32 bits asignada a los anfitriones que utiliza TCP/IP. Una dirección IP pertenece a una de las cinco clases (A, B, C, D y E) y se escribe como cuatro bytes separados con puntos. Cada dirección consiste en un número de red, un número de subred opcional y un número de host.

- ✓ **DNS:** Sistema de Nombres de Dominio. Sistema utilizado en Internet para traducir los nombres de los nodos de red en direcciones.

- ✓ **FQDN:** Fully Qualified Domain Name. Es un nombre entendible por personas que incluye el nombre de la computadora y el nombre de dominio asociado a la misma. La longitud máxima permitida para un FQDN es 255 bytes, con una restricción adicional a 63 bytes por etiqueta dentro de un nombre de dominio.

- ✓ **FTP:** Protocolo de Transferencia de Archivos. Protocolo de aplicación, parte de la pila de protocolos TCP/IP, que se utilizan para la transferencia de archivos entre los nodos de la red.

- ✓ **GPL:** Licencia Pública General. Es una licencia creada por la Free Software Foundation, está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

- ✓ **Groupware:** Se refiere a los programas informáticos que integran el trabajo de un proyecto con muchos usuarios concurrentes que se encuentran en diversas estaciones de trabajo, típicamente conectados a través de una red Internet o de una intranet.

- ✓ **HTML:** Lenguaje de Marcado de Hipertexto. Lenguaje de formateo de documentos simples en hipertexto, que utiliza etiquetas para indicar como debe ser interpretada determinada parte de un documento por una aplicación de visualización, como un navegador WWW.

- ✓ **HTTP:** Protocolo de Transferencia de Hipertexto. Especifica cuales mensajes pueden enviar los clientes a los servidores y que respuestas obtienen. Todos los clientes y servidores deben obedecer este protocolo.

- ✓ **IMAP:** Protocolo de Acceso a Mensajes de Internet. IMAP proporciona mecanismos de gran alcance para leer mensajes o incluso partes de un mensaje. Debido a que la suposición más razonable es que los mensajes no se transferirán a la PC del usuario, IMAP proporciona mecanismos para crear, destruir y manipular múltiples buzones en el servidor.

- ✓ **IP:** Protocolo de Internet. Protocolo de la capa de red en la pila TCP/IP que ofrece un servicio de red sin conexión. El protocolo IP proporciona características de direccionamiento, especificación de tipo de servicio, fragmentación, reensamblado y seguridad.

- ✓ **JDBC:** Java Data Base Connectivity. Es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

- ✓ **JVM:** Maquina Virtual de Java. Es un lenguaje de maquina orientado a pilas.

- ✓ **LANs:** Redes de Área Local. Red de alta velocidad y baja tasa de errores, que cubre un área geográfica relativamente pequeña (hasta algunos miles de metros). Las LANs conectan estaciones de trabajos, periféricos, terminales y otros dispositivos en un solo edificio u otra área geográfica limitada.

- ✓ **LDA:** Agente de Entrega Local.

- ✓ **LDAP:** Protocolo Ligero de Acceso al Directorio. Es un conjunto de protocolos diseñados para acceder a los directorios de información y mantenerlos.

- ✓ **LMTP:** Protocolo de Transferencia Local de Correo. Proporciona los mecanismos necesarios para la transporte de correo fiable y eficaz.

- ✓ **MDA:** Agente de Entrega de Correo. Utilizado por los agentes MTA para entregar correo electrónico al buzón de un usuario concreto.

- ✓ **MTA:** Agente de Transferencia de Correo. Su función consiste en aceptar correo electrónico para sus clientes y almacenarlos en buzones de una maquina ISP.

- ✓ **MUA:** Agente de Correo de Usuario. Programa que permite al usuario leer y escribir correo electrónico.

- ✓ **OSI:** Interconexión de Sistemas Abiertos. Es el programa de estandarización internacional creado por la ISO y la ITU-T para desarrollar estándares para las redes de datos que faciliten la interoperabilidad de equipos fabricados por diferentes proveedores.

- ✓ **OX:** Open-Xchange. Es una aplicación para el trabajo en equipo o sistema colaborativo que proporciona a sus usuarios un avanzado sistema de comunicaciones y funciones para la colaboración.

- ✓ **PAM:** Modulos de Autenticación Conectables. Se trata de un mecanismo que proporciona una interfaz entre las aplicaciones de usuario y diferentes métodos de autenticación, tratando de esta forma de solucionar uno de los problemas clásicos de la autenticación de usuarios.

- ✓ **PC:** Computador Personal.

- ✓ **PHP:** Preprocesador de Hipertexto. Especialmente bueno para manejar formularios.

- ✓ **POP:** Protocolo de Oficina de Correo. Este protocolo permite que los agentes de transferencia de usuario (en PCs clientes) contacten al agente de transferencia de mensajes (en la maquina del ISP) y que el correo electrónico se copie desde el ISP al usuario. El puerto TCP utilizado es el 110.

- ✓ **RBL:** Listas Negras de Filtrado Real. Utilizadas para verificar dominios o direcciones de correo utilizadas para enviar SPAM.

- ✓ **Servidor:** (Server) Es un nodo o programa de software que provee servicios a clientes.

- ✓ **SGBD:** Sistema Gestor de Bases de Datos. Es un conjunto de programas que permiten crear, manipular y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad.

- ✓ **SMTP:** Protocolo de Transferencia Simple de Correo. Es el estándar de Internet para el intercambio de correo electrónico.

- ✓ **SMTP:** Protocolo Simple de Transporte de Correo. Es un protocolo de Internet que proporciona servicios de correo electrónico.

- ✓ **Solicitud (query):** Mensaje para solicitar información sobre el valor de alguna variable o conjunto de variables.

- ✓ **SSH:** Seguridad de Consola. Protocolo que sirve para acceder a máquinas remotas a través de una red.

- ✓ **SSL:** Capa de Sockets Seguros. SSL construye una conexión segura entre los dos sockets, incluyendo negociación de parámetros entre el cliente y el servidor, autenticación tanto del cliente como del servidor, comunicación secreta y protección de la integridad de los datos.

- ✓ **TCL:** del acrónimo en inglés "**T**ool **C**ommand **L**anguage" o lenguaje de herramientas de comando, actualmente se escribe como "Tcl" en lugar de "TCL", es un lenguaje de script creado por John Ousterhout, que ha sido concebido para su fácil aprendizaje, pero que resulta muy potente en las manos adecuadas. Se usa principalmente para el desarrollo rápido de prototipos, aplicaciones "script", interfaces gráficas y pruebas.

- ✓ **TCP:** Protocolo de Control de Transmisión. Protocolo orientado a la conexión que pertenece a la capa de transporte y que ofrece una trasmisión confiable de datos duplex total. TCP es parte de la pila de protocolos de TCP/IP.

- ✓ **TLS:** Seguridad de la Capa de Transporte.

- ✓ **URL:** Localizador Uniforme de Recursos. La URL sirve efectivamente como nombre mundial de páginas Web. Tienen tres partes: el protocolo, el nombre DNS donde se encuentra la página y un nombre local que indica de manera única la página específica.

- ✓ **W3C:** World Wide Web Consortium.

WHOIS: Es un protocolo TCP basado en preguntas/repuestas que es usado para consultar de una base de datos para determinar el propietario de un nombre de dominio o una dirección IP en Internet.

XIII. BIBLIOGRAFÍA.

- ✓ http://bulma.net/body.phtml?nIdNoticia=2361
- ✓ http://en.wikipedia.org/wiki/Open-Xchange
- ✓ http://es.tldp.org/Manuales-LuCAS/doc-tutorial-postfix-ldap-courier-spamassassin-amavis-squirrelmail/html-multiple/
- ✓ http://es.wikipedia.org/wiki/Open-Xchange
- ✓ http://gpl.netixia.com/openxchange/openxchange-sarge-howto.html
- ✓ http://opensuse.cict.fr/distribution/SL-10.1/inst-source/docu/es/reference_es.pdf
- ✓ http://open-xchange.org/oxwiki/OXDebianSargeFromPackage
- ✓ http://www.asg.web.cmu.edu/cyrus/
- ✓ http://www.heinous.org/wiki/Open-XChange_on_SUSE_10.1
- ✓ http://www.liberaliatempus.com/articulos/linux/
- ✓ http://www.mindes.gob.pe/manual_ox
- ✓ http://www.openldap.org/
- ✓ http://www.open-xchange.com/
- ✓ http://www.open-xchange.org
- ✓ http://www.open-xchange.org/oxwiki/
- ✓ http://www.open-xchange.org/oxwiki/Quick_20Administrative_20Scripts
- ✓ http://www.postfix.org/
- ✓ http://www.x-tend.be/~raskas/openxchange/
- ✓ Redes de Computadoras. Cuarta Edición. Andrew S. Tanenbaum.