ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

"HERRAMIENTA DE ADMINISTRACION PARA EL SYSLOG DE UNIX"

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO EN COMPUTACIÓN

ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

ESPECIALIZACIÓN SISTEMAS MULTIMEDIA

Presentado por:

ORLANDO CRESPO LEON

JUAN MARIN RAMALLO

GUAYAQUIL - ECUADOR

2008

AGRADECIMIENTO

A Dios por ayudarnos siempre y en cada momento de nuestras vidas.

A la ESPOL y sus profesores por inculcarnos conocimientos éticos y profesionales a lo largo de la carrera.

A nuestra directora de tesis la Ing. Cristina Abad por brindarnos su apoyo, tiempo y conocimientos para el desarrollo de esta tesis.

A nuestros padres por ser siempre nuestro apoyo y soporte y por brindarnos la oportunidad de poder estudiar y ser profesionales en nuestras vidas.

A nuestros familiares y amigos, los cuales demostraron su preocupación y apoyo en este largo camino.

DEDICATORIA

A Dios.

A nuestras familias por ser el apoyo incondicional en cada paso dado en nuestras vidas.

A nuestros seres queridos.

TRIBUNAL DE GRADUACIÓN

Ing. Holger Cevallos

SUB-DECANO DE LA FIEC

Ing. Cristina Abad

DIRECTORA DE TESIS

Ing. Carmen Vaca

MIEMBRO PRINCIPAL

Ing. Guido Caicedo

MIEMBRO PRINCIPAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de Graduación de la ESPOL).

Orlando Crespo León

Juan Marín Ramallo

RESUMEN

Una de las tareas más importantes para los administradores de Unix es llevar el control de todas las funcionalidades del sistema. Por esta razón el uso de bitácoras o logs puede ayudar significativamente a los administradores en el control y prevención de errores en el sistema.

Existen muchas herramientas en el mercado que ayudan a los administradores a llevar un control del funcionamiento del sistema, pero la mayoría de ellas son para usuarios expertos, los cuales fácilmente pueden identificar los errores con solo leer la línea de error. Para un usuario inexperto, esto puede resultar una tarea muy complicada. Además la mayoría de estos productos suelen ser demasiado costosos e incluso difíciles de operar, ya que se basan en un solo tipo de usuario.

Esta tesis propone el desarrollo de un Sistema de Administración de Logs en Unix, que permita a los diferentes tipos de usuarios (expertos e inexpertos) llevar un correcto control de los sistemas Unix, por medio de estadísticas y reportes de errores los cuales ayudarán a los usuarios a visualizar de una manera más fácil los errores encontrados tanto de forma local como remota.

El sistema consta de una serie de alarmas que permitirán al usuario controlar errores de tipo *emerg*, *critic* (según terminología del syslog de Unix) los cuales pueden afectar de manera severa el correcto funcionamiento del sistema.

En el Capítulo 1 se explica la importancia de administrar los logs, los problemas que estos representan y las posibles soluciones y beneficios de administrarlos correctamente.

En el Capítulo 2 se da una breve explicación de administración y seguridad de los logs en Unix, se establece una comparación entre las herramientas existentes en el mercado, una breve explicación de los sistemas operativos Unix existentes (donde se probará el sistema) y una descripción de los posibles usuarios del sistema.

En el Capítulo 3 se analizan los requerimientos del sistema para la instalación de los diversos sistemas Unix, los requerimientos funcionales y no funcionales, las tecnologías empleadas. Aquí también se describe los casos de uso y escenarios del sistema, detalles de la interacción Usuario-Sistema, y herramientas utilizadas para el desarrollo mismo.

En el Capítulo 4 se describe las características relacionadas con el diseño de la arquitectura del sistema, la base de datos e interfaz del usuario a fin de cumplir con los requerimientos iniciales, descripción de entidades, diagrama entidad-relación, flujo de ventanas y el diseño de la interacción del usuario.

En el Capítulo 5 se desarrolla un plan de pruebas del correcto funcionamiento del sistema, explorando todos los casos posibles y un breve detalle del proceso de implementación.

Finalmente se indican conclusiones y recomendaciones obtenidas en el desarrollo de la tesis, y se incluyen los apéndices.

ÍNDICE GENERAL

INDICE DE ABREVIATURAS

BCPL: Basic Combined Programming Language.

BSD: Berkeley Software Distribution.

DNS: Domain Name System.

FIFO: First In, First Out.

FTP: File Transfer Protocol.

GNU: GNU is Not Unix.

HTML: Hypertext Markup Language.

IHM: Interacción Hombre Máquina.

IMAP: Internet Message Access Protocol.

IRC: Internet Relay Chat.

SMTP: Simple Mail Transfer Protocol.

SSH: Secure Shell.

TCP: Transmission Control Protocol.

UDP: User Datagram Protocol.

XML: Extended Markup Language.

INDICE DE FIGURAS

INDICE DE TABLAS

INTRODUCCIÓN

Un log o bitácora digital es un archivo que registra movimientos y actividades de un determinado programa y mantiene gran variedad de información desde diagnósticos de utilización de recursos a problemas relacionados con el hardware o software, problemas de seguridad o pistas de intrusión.

El presente proyecto de tesis tiene como objetivo el desarrollo de una herramienta que controle los logs de Unix, mediante el uso de métodos como la visualización directa de los logs, estadísticas, reportes y consultas, facilite la administración y seguridad de los eventos que generan las diferentes aplicaciones y tareas que se encuentran en ejecución tanto por parte de los usuarios como del sistema operativo.

La administración correcta de los logs, proporciona estabilidad y seguridad al sistema mediante la temprana detección de errores y fallas eventuales, dando al usuario administrador la posibilidad de controlar estos problemas a través de una serie de **niveles de error** (*debug, info, warning,* etc.,) ya sea de forma local o remota, y de comprender el significado de mensajes complejos generados tanto para los usuarios expertos como inexpertos.

La solución desarrollada para la demostración de este proyecto de tesis, consiste en una herramienta que funciona bajo la plataforma Unix y puede trabajar sin ningún inconveniente bajo cualquier versión del kernel de Linux, los cuales poseen similares características de funcionamiento.

CAPÍTULO 1

ANTECEDENTES Y JUSTIFICACIÓN

1.1. La importancia de llevar un control sobre el Registro de Eventos en los sistemas actuales.

Muchas son las aplicaciones, dispositivos y sistemas que a diario hacen uso de los registros de eventos, comúnmente conocidos como logs o bitácoras, los que permiten al usuario administrador determinar el estado del sistema y ejecutar acciones correctivas o preventivas en base al análisis de éstos.

Según el diccionario Oxford, un log es un registro de eventos en un tiempo particular, para registrar datos o información sobre quién, qué, cuándo, dónde y por qué un evento ocurre para un dispositivo en particular o aplicación.

Aún cuando los logs, como lo indica su definición, anotan todos los eventos ocurridos, su importancia se ha incrementado por cuanto constituye un medio para prevenir un daño fatal al sistema, los cuales pueden evitarse mediante la exploración temprana de la integridad de los mensajes registrados, que alertarán al administrador a levantar defensas contra ataques. Esto es posible, puesto que mediante un log se puede llevar un control cronológico de los eventos que suceden en el sistema indicando así, cuantas veces un error está ocurriendo y si éste puede o no causar problemas futuros.

Los logs permiten ejercer un control sobre el acceso de usuarios al sistema y un monitoreo de sus acciones, aportando mayor seguridad. La detección y prevención temprana de un ataque de personas no autorizadas puede marcar la diferencia entre el éxito o fracaso de una organización, por cuanto modificaciones de intrusos sobre información clasificada podría ocasionar daños irreparables con consecuentes pérdidas millonarias.

1.2. Problemática.

Los eventos que ocurren dentro de un sistema proceden de distintas fuentes (red, aplicaciones, usuarios, unidades, etc.), por lo cual la información registrada en un log es de lo más diversa, la cual además varía de acuerdo al sistema operativo en el que nos encontremos.

La tarea de administrar un log suele ser tediosa y complicada, orientada hacia el usuario experto que posea la habilidad de descifrar los diferentes parámetros que se describen en ellos.

Una particularidad común a todos los archivos de registro de eventos sin importar el S.O donde residan, es que *crecen*, lo que obliga a definir alguna política para el manejo de estos archivos a fin de evitar que llenen espacio en disco [1].

Al referirnos al sistema operativo Unix[1] sobre el cual se centra esta tesis, las dificultades inminentes a la administración e interpretación de logs, se ven acrecentadas por la muy conocida falta de relación entre los comandos propios del sistema operativo y su

[1] En esta tesis, el término Unix hace referencia a los sistemas tipo Unix o basados en Unix, como GNU/Linux, BSD, etc., y no solamente a aquellos que cumplen con los requisitos para llamarse Unix®.

funcionalidad, lo cual se manifiesta también por medio de los términos utilizados en los logs. Esta carencia de familiaridad, ocasionará que la tarea de administrar sea más complicada para usuarios inexpertos y poco habituados a este ambiente.

Aún cuando la base para el almacenamiento de logs es la misma para todas las distribuciones basadas en Unix, el control del registro de eventos puede variar, dependiendo de la configuración del sistema. Esta variación se manifiesta por la ubicación de los archivos resultantes de los logs, lo que dificulta más aún al administrador la tarea de localizar el log asociado a un dispositivo o aplicación.

1.3. Alternativas de Solución.

La tarea de administrar, requiere de una gran capacidad de análisis e interpretación a fin de determinar qué eventos pueden constituir un eventual riesgo o hueco para el sistema.

El administrador además de realizar sus actividades cotidianas como otorgar permisos, crear nuevos usuarios u otras, debe lidiar con cientos de archivos logs generados para distintos aplicativos o tareas que se ejecutan en el S.O. Cada archivo a su vez puede contener cientos e incluso miles de eventos registrados, por lo cual se debe disponer de algún mecanismo que permita una revisión rápida y eficaz de los mismos, sin la necesidad de revisarlos uno a uno.

Los mecanismos de revisión pueden variar desde aquellos con interfaz gráfica que ofrecen un visor, a aquellos que requieren la ejecución de comandos desde consola tales como `vi`, `grep`, `find`, `more`, `cat`, `head`, `last`, y `finger`.

Sin embargo, estas soluciones suelen estar limitadas a una sola distribución de Unix o a una ubicación específica. Al no ofrecer la apertura necesaria para indicar distintas rutas posibles para almacenar los logs, limitan su alcance produciendo un control incompleto del sistema.

Por lo indicado, se demuestra que la tarea de administrar no es fácil, pues involucra la revisión cuidadosa de cientos de líneas de mensajes generados. Cuando quien maneja el sistema es un usuario inexperto, esta tarea puede volverse aún más compleja; pues primero deberá conocer y familiarizarse con los comandos, y luego con la administración propia de los logs para la versión de la distribución con la cual trabaja.

1.4. Descripción General de la Solución.

Esta tesis propone el desarrollo de un **Sistema de Administración de Logs para Unix**, que permita a los diferentes tipos de usuarios (expertos e inexpertos) llevar un correcto control de los eventos registrados, por medio de la generación de reportes y estadísticas.

La solución busca proporcionar una administración basada en configuraciones propias del syslog de Unix, donde el usuario independientemente de su nivel de pericia con el sistema operativo o distribución, pueda elegir entre una configuración original y personalizada, y en la que también pueda precisar qué información desea le sea notificada o alertada para establecer un control anticipado.

El sistema admite el establecimiento de listas de exclusión de mensajes que el administrador considere eludibles o innecesarias de reportar, evitando así que el usuario visualice mensajes de logs innecesarios.

El sistema posee también vistas de los archivos de registro de eventos, con filtros que permiten depurar el tipo de notificación a mostrar por medio de diversos parámetros tales como: fecha, demonio, nivel de error y mensaje.

Adicionalmente, el sistema consta de alarmas que permitirán al usuario controlar errores de tipo *emerg* o *critic;* los cuales pueden afectar de manera severa el correcto funcionamiento del sistema. El sistema permite también el envío de correos electrónicos al administrador, mensajes al celular o notificaciones visuales, para advertir la presencia de mensajes *critic* o *emerg*, característica que dependerá del usuario utilizarla o no.

También permite generar respaldos de los archivos de log, mediante filtros tales como la fecha, servicio y nivel.

1.5. Beneficios

Beneficios de los Registros de Eventos

Los beneficios asociados a los registros de eventos pueden señalarse con respecto a una parte del sistema operativo o a su totalidad.

Entre los más significativos podemos anotar:

- Permite identificar problemas asociados a una aplicación o dispositivo. Esta información se puede utilizar para decidir dónde es necesario llevar un control y qué acciones se deben tomar ante estos eventos.
- Permite determinar la ocurrencia de un evento fallido al revisar cronológicamente su aparición. Esto se debe a que los archivos logs conservan un registro por fecha y hora de cada evento ocurrido en el sistema.

- A través de los registros de eventos se puede encontrar información para detectar posibles problemas asociados al sistema como una falla, o una incidencia de seguridad.

Beneficios de la Solución Implementada

El beneficio principal de la solución consiste en la configuración del archivo *syslog.conf*, de forma transparente para el usuario administrador y en términos entendibles que lo hacen independiente del nivel de conocimiento o destreza del usuario en ambientes Unix.

Otros beneficios adicionales son:

- Habilita notificaciones sobre cada uno de los demonios (servicios residentes en memoria), donde el administrador podrá configurar el destino de cada registro de eventos, pudiendo ser a un correo electrónico o hacia un archivo de texto.

- Cada log es mostrado al usuario y cuenta además con filtros basados en las características de los eventos (fecha, nivel de error, demonio y mensaje), que el usuario podrá visualizar según su criterio, teniendo además la oportunidad de exportar el resultado en formato XML[2].

- Las alarmas, para el caso de los mensajes con nivel de error *critic* y *emerg*, pueden ser sonoras o visuales, ejecutadas en tiempo real cuando se genere este tipo de error.

- Los reportes y estadísticas pueden ser sencillos o avanzados, basándose en los filtros que el usuario considere conveniente por cada una de las columnas de un log (fecha,

---

[2] El Extensible Markup Language (XML) es un metalenguaje (*lenguaje usado para hacer referencia a otros lenguajes*) extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

máquina, demonio, mensaje). Estos reportes facilitan al usuario la toma de decisiones sobre el funcionamiento del sistema.

- Dado que un registro de eventos provee abundante información del sistema, los intrusos podrían emplear estos registros para obtener información confidencial o restringida convirtiéndolo en un punto vulnerable de ataques. Por esta razón el sistema extrae los mensajes de logs del sistema y los guarda en la base de datos directamente; cuando se sale del sistema, estos mensajes se borran automáticamente.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Administración y Seguridad en los logs Unix.

Los logs registran los eventos que ocurren en el sistema y son generados por procesos que se ejecutan en el S.O. Estos eventos pueden ser accesos al sistema, mensajes de información del kernel, el tráfico capturado por un analizador de tráfico (sniffer) e incluso los famosos logs del IRC[3] [2].

Su utilidad suele estar enfocada primordialmente en que constituyen herramientas útiles para la localización y resolución de problemas, y para reconocer actividades inapropiadas por medio de la determinación de patrones que pudieran advertir sobre una posible infiltración, virus o fallas del sistema.

Las versiones de Unix están pre-configuradas para registrar cierta información en los archivos de log, pero se pueden establecer configuraciones adicionales a criterio del usuario administrador para aumentar aún más la información registrada.

---

[3] IRC (Internet Relay Chat), protocolo de comunicación en tiempo real basado en texto, que permite debates en grupo o entre dos personas y que está clasificado dentro de la mensajería instantánea.

Un administrador que examine regularmente los logs aprenderá mucho sobre cómo funciona el sistema, pudiendo garantizar menos caídas y al mismo tiempo notará cuándo ocurran brechas de seguridad, especialmente si se utilizan alertas.

Existen dos tipos de archivos logs: los de texto plano y los binarios. La información de éstos últimos, sólo es visible mediante el uso de aplicaciones externas. En los sistemas Unix, la mayor parte de información de registro es mantenida en el directorio /var.

Las tablas I y II, indican las rutas de los registros más frecuentes en ambientes Unix, así como la descripción de la información que almacenan.

ARCHIVOS DE LOG BINARIOS		
Ruta	**Descripción**	**Comando**
/var/log/wtmp	Registra las conexiones y desconexiones al sistema.	last
/var/log/utmp	Registra la información de usuarios conectados, exceptuando a aquellos que emplean servicios como FTP, IMAP, etc.	who
/var/log/lastlog	Registra la última conexión (fecha y hora) de cada usuario en el sistema.	lastlog
/var/log/faillog	Registra intentos fallidos de conexión fallidos.	faillog

TABLA I. Archivos de Log Binarios.

La tabla describe la ruta de los logs binarios más comunes, junto con la descripción de la información que registran y el comando empleado para visualizarla.

ARCHIVOS DE LOG DE TEXTO PLANO

Ruta	Descripción
/var/log/Syslog	Almacena todos los logs por defecto. Si la configuración original no ha cambiado, estarán registrados la mayoría de eventos.
/var/log/messages	Registro de mensajes del sistema (informativo).
/var/log/secure	Registro de las conexiones realizadas hacia la máquina.
/var/log/auth.log	Registro de las autenticaciones ocurridas en el sistema, tanto las aceptadas como las fallidas.
/var/log/debug	Registro de depuración de los programas ejecutados en el sistema, enviada por los programas o el kernel del S.O
/var/log/kern.log	Mensajes procedentes del kernel
/var/log/daemon.log	Logs de demonios ejecutados en el S.O.
/var/log/mail.log	Registro de mensajes entrantes o salientes del servidor SMTP[4] del equipo.
/var/log/boot.log	Mensajes de arranque del sistema.
/var/log/sulog	Registros de ejecuciones del comando su[5].

TABLA II. Archivos de Log de Texto Planos.

La tabla describe la ruta de los logs de texto plano más comunes, junto con la descripción de la información que registran.

El protocolo syslog, definido en el RFC3164 [29], fue originalmente escrito por Eric Allman. Este protocolo provee un mecanismo para permitir a un dispositivo enviar mensajes de notificación de eventos a través de redes IP hacia colectores de mensajes de eventos,

[4] Simple Mail Transfer Protocol (SMTP), o protocolo simple de transferencia de correo.

[5] Comando de UNIX que permite emplear el intérprete de comandos de otro usuario sin necesidad de cerrar la sesión

también conocidos como servidores syslog. El protocolo está diseñado sencillamente para transportar estos mensajes de eventos desde el dispositivo generador hacia el dispositivo colector. El colector no devuelve un reconocimiento de la recepción de mensajes En un sistema operativo Unix, el kernel y otros componentes internos generan mensajes y alertas. Estos mensajes son típicamente almacenados en un sistema de archivos o retransmitidos hacia otro dispositivo en la forma de mensajes syslog [3].

El demonio interno de Unix, llamado *syslogd*, proporciona el servicio necesario para controlar el syslog. Este demonio es una parte integral de la mayor parte de distribuciones Unix, y no requiere ser descargado o instalado. El syslog provee un punto central para recopilar y procesar los logs del sistema, donde todos los eventos que se deseen controlar en el sistema son registrados en el archivo de configuración `/etc/syslog.conf`.

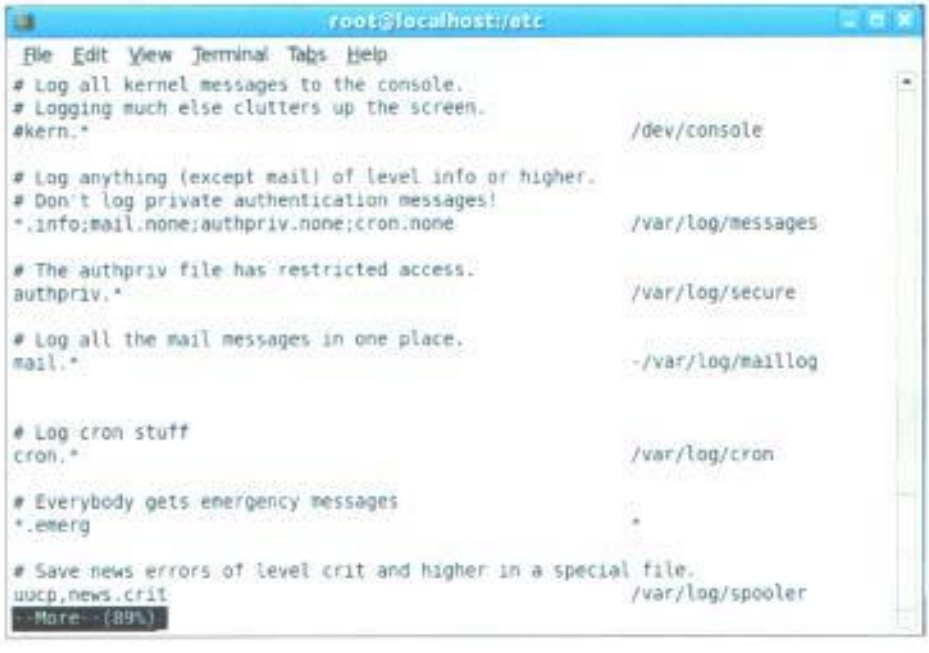

Figura 1. Archivo *syslog.conf* con rutas de almacenamiento.
La imagen muestra la estructura del archivo *syslog.conf* en Fedora Core 7.

Como lo indica la Figura 1, en el archivo *syslog.conf* se encuentran líneas semejantes a:

```
*.info;*.notice /var/log/messages
```

Cada entrada del archivo posee tres elementos:

1. El primer elemento es la fuente de la información de log o servicio, llamada *facility* – en otras palabras qué procesos deberían monitorearse. Los servicios son: `auth`, `authpriv`, `cron`, `daemon`, `kern`, `lpr`, `mail`, `mark`, `news`, `syslog`, `user`, `uucp`, `local0` a `local7`.

   Si se indica *, se hace referencia a todas las fuentes de log disponibles en el sistema.
2. La información que le sigue especifica la clase de eventos que serán capturados, conocida como prioridad, los cuales pueden ser: `debug`, `info`, `notice`, `warn`, `err`, `crit`, `alert`, `emerg`. Por ejemplo, si se indica `*.info`, todas las fuentes de log serán monitoreadas y sólo los eventos con prioridad `info` serán capturados.
3. El último elemento especifica lo que se hará con los eventos registrados, es decir, dónde será almacenada la información resultante, pudiendo ser un archivo, un terminal remoto, un usuario, una impresora, un archivo FIFO u otro host.

   En el ejemplo, el resultado será almacenado en el archivo `/var/log/messages`.

El *facility* y la *prioridad* en conjunto se conocen como selector y el destino como la acción. Las tablas III y IV resumen los facilities más comunes y las prioridades.

Facility	Descripción
auth/security	Registra información relacionada con la autenticación de usuarios.
authpriv	Versión más detallada de auth.
cron	Registra la actividad del demonio cron, empleado para planificar eventos.
daemon	Recopila información sobre todos los demonios del sistema.
mail	Rastrea las actividades del servidor de correo.

syslog	Información acerca del servidor syslog.

TABLA III. Facilities más comunes.

Prioridad	Uso
info	Registra mensajes de información general.
notice	Registra ocurrencias que requieren interacción especial
warning	Advertencia del sistema, "puede haber un problema".
err	Error en el sistema, "un problema ocurrió"
crit	Condición crítica, "a gran problema ha ocurrido"
alert	Se requiere intervención, "un problema está ocurriendo, necesita ayuda"
emerg	Seria Falla del Sistema
panic	Sistema en peligro "GRAN problema"
debug	Registra casi todo.

TABLA IV. Uso de prioridades.

Además de conocer el funcionamiento del syslog, un administrador en Unix debe considerar tareas tales como el almacenamiento de los registros de log y la distribución de los mismos. Al hacerlo podría aplicar distintas políticas como centralizar la administración del *syslog.conf* a un servidor, y distribuir el almacenamiento de cada tipo de log en distintos servidores ubicados local o remotamente.

El administrador no puede deshacerse de la información registrada, ya que constituye prueba sustancial en caso de un posible ataque, por lo cual requerirá establecer respaldos

periódicos hacia dispositivos externos tales como cintas magnéticas u otro medio de almacenamiento.

Para el propósito de añadir seguridad, debería considerarse encriptar la información de registros.

Otro punto a considerar en administración de logs, es la rotación, la cual consiste en almacenar los logs en otros archivos cada vez que estos cubran el espacio designado en disco. Para tal fin existen utilidades que pueden ser instaladas, siendo *logrotate*[6] una de ellas.

2.2 Comparación entre Herramientas existentes para la administración de logs en Unix.

Logsurfer [4]

Características

Herramienta de monitoreo a tiempo real que tiene la capacidad de crear reglas dinámicas, está desarrollado en lenguaje C y bajo proceso de carga en los equipos donde se instale, puede llamar a programas externos y necesita pocas líneas de configuración para activar las alarmas de seguridad, sus búsquedas son a través de expresiones regulares.

Desventajas

No posee ninguna seguridad con los archivos logs generados, lo que permitiría a los intrusos leer información importante. Además, no posee reportes ni estadísticas que permitan a los administradores tomar decisiones. Se encuentra disponible sólo en inglés.

---

[6] Programa que permite rotar, comprimir, eliminar y enviar automáticamente por correo los archivos de registro.

Logwatch [5][6]

Características

Logwatch es un sistema de análisis de logs personalizable, que realiza un análisis a través del sistema de logs por un periodo dado de tiempo y genera un reporte analizando áreas que el usuario especifica. Es fácil de usar y trabaja sin problemas en muchos sistemas. Se ejecuta cada noche y envía un mail indicando los resultados encontrados. Puede también ser ejecutado desde la línea de comando.

La salida es por servicio y el usuario puede limitar la salida a un servicio particular. Los subscripts, que son responsables para generar una salida entendible para el usuario, comúnmente convierten las líneas de logs generadas por defecto en un formato estructurado entendible.

Desventajas

Logwatch generalmente ignora el componente tiempo en los detalles del log, el usuario sabría que el evento reportado fue detectado en un rango de tiempo solicitado, pero debería acudir a los datos del log original para conocer más información acerca del registro. Además, puede generar información no necesaria para el administrador, dado que la salida es siempre por servicio y no por servicio-prioridad.

Logcheck [7][8]

Características

Logcheck es lanzado por el cron cada cierto tiempo y mediante comparaciones. Trata los datos almacenados en los logs contra una serie de reglas (expresadas en términos de expresiones regulares).

Revisa periódicamente los logs del sistema, analizando cada una de las líneas y clasificándolas según diferentes niveles de alerta; reportándolo al administrador del sistema

en un formato fácil de leer, descartando las líneas que no tengan relevancia, y enviándolo por correo. Logcheck revisa cada línea contra cuatro niveles de seguridad: ignorar, actividad inusual, violación de seguridad y ataque.

Desventajas

Dado que realiza revisiones periódicas, si un mensaje tipo *critic* o *alert* surgiera, el tiempo de aviso podría afectar la rápida detección del problema. La configuración errónea de alguno de los 4 archivos a administrar puede elevar el número de mensajes generados. Tampoco provee una solución de seguridad contra intrusos.

Swatch [8][9]

Características:

Es usado para monitorear archivos de logs. Cuando una línea de log concuerda con un patrón especificado por el usuario, ésta se marca y se genera una salida de texto o un programa externo notifica al usuario a través del correo o algún otro medio disponible.

Swatch no sólo escanea periódicamente los archivos de logs y le dice al usuario lo que ha sucedido, sino que también puede activamente explorar las entradas de los archivos logs generadas por *syslogd* y decir qué ocurre en ese momento. Adicionalmente toma acciones referentes al problema encontrado al momento de explorar los logs.

Desventajas

El problema de swatch es que lleva el control de un solo log a la vez. Para esto, el usuario deberá en un solo archivo de log registrar todo lo que desea monitorear, lo que puede resultar una tarea compleja. Además no existe ninguna seguridad para los archivos log generados permitiendo a los intrusos poder visualizar información restringida.

Necesita tener instalado Perl y posee reglas dinámicas básicas para la exploración de logs (puede ignorar mensajes repetidos si el archivo de log se escanea cada cierto tiempo).

Psionic Logcheck [10][11]

Características

Psionic Logcheck analiza los archivos de log a intervalos regulares de tiempo vía `crontab`, y envía un correo al administrador indicándole todos los problemas encontrados. Es fácilmente configurable, se modifican dos archivos donde se indica qué se permite y qué no al momento del análisis de los logs.

Posee alarmas que pueden activarse frente a la detección de intrusos o por fallas de algún dispositivo o programa.

Desventajas

Requiere la configuración de dos archivos: `host.allow` y `host.deny`, lo que además de constituir una tarea tediosa, podría resultar compleja si el usuario no posee la experiencia adecuada.

Colorlogs [12]

Características

Colorlogs utiliza un mecanismo de coloreo de archivos logs, basándose en la configuración establecida. Busca palabras claves y colorea las líneas con las coincidencias encontradas.

Desventajas:

Monitorea un log a la vez empleando la configuración original del sistema, lo cual restringe al usuario. No genera reportes ni alarmas.

2.3 Posibles usuarios del sistema

El presente proyecto de tesis, está orientado para el siguiente grupo de usuarios:

- **Administradores de Unix:** Basados en la información obtenida de los logs pueden administrar mejor los recursos, llevar un control monitoreado de los logs y realizar una detección temprana de los posibles problemas.

- **Administradores de seguridad:** Basados en la información de los logs, pueden detectar problemas relacionados con intrusos en el sistema, establecer políticas de seguridad y realizar análisis forense[7].

- **Personas que administran ambientes tanto en Windows como en Unix/Linux**: obtendrían beneficios de aprender cómo integrar los registros de Windows en un ambiente de autenticación en Unix.

- **Personas con un sistema operativo basado en Unix:** Con conocimientos básicos de Unix y de sus comandos esenciales, pueden emplear la información basada en los logs para evitar posibles problemas en el funcionamiento de su sistema.

[7] Obtención y análisis de datos empleando métodos que distorsionen lo menos posible la información con el objetivo de reconstruir todos los datos y/o los eventos qué ocurrieron sobre un sistema en el pasado.

CAPÍTULO 3

ANÁLISIS DEL SISTEMA

3.1 Análisis de requerimientos y alcance del sistema

Para simplificar la tarea de administrar un sistema basado en Unix, se ha creado esta herramienta que permite llevar un control del estado del sistema a partir de los archivos de logs.

A continuación se detallarán los requerimientos funcionales y no funcionales referentes al desarrollo de este sistema.

3.1.1. Requerimientos funcionales

Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Esta tesis provee una interfaz que permite a los administradores de sistemas Unix, llevar un control de todas las tareas del sistema, sean éstas de correo, impresión, usuarios, demonios, etc. mediante búsquedas, reportes, estadísticas y alarmas, con el objetivo de facilitar la tarea de detección de errores a los administradores y prever posibles fallas futuras.

3.1.2. Especificaciones de funcionalidad

Configuración del *syslog.conf*: Siendo el *syslog.conf* el archivo donde se detalla la configuración que tomará el sistema para administrar y almacenar los mensajes de logs, se ha dispuesto una interfaz que permitan al administrador, llevar un mejor manejo de la configuración de modo entendible para el usuario, tratando de evitar términos que resultaren complejos para cualquier tipo de usuario que emplee el sistema.

Las opciones disponibles son las siguientes:

- **Configuración Original:** La configuración por defecto indicada en el archivo *syslog.conf* es empleada para la administración el sistema.

- **Configuración Personalizada:** Permite al usuario elegir qué es lo que desea administrar, para ello dispondrá de una interfaz que permita seleccionar el servicio y la prioridad que desea sea mostrada en el visualizador de logs.

Vistas de los archivos logs: Se disponen de vistas directas de los registros de eventos en columnas, donde el usuario podrá ordenar los resultados por cualquiera de los campos de visualización.

Además existe la posibilidad de filtrar por mensaje, fecha, demonio dentro de la misma ventana de vistas de eventos. También se permite al usuario exportar estos archivos del registro de eventos en formato XML para su uso personal.

Reportes: Los reportes serán filtros sobre los archivos log, por cualquiera de sus campos: fecha, hora, máquina, demonio y mensaje. El usuario tendrá la posibilidad de filtrar los archivos de logs por uno o varios campos y además exportar el resultado como archivo XML.

Estadisticas del sistema: Se basa en filtros por fecha o por servicio/prioridad pero de manera gráfica, a través de líneas de tiempo, histogramas y gráfico circular.

Alarmas del sistema: Estas alarmas brindan al usuario la posibilidad de ser avisado cuando un error de prioridad *critic* o *emerg* ha surgido. Estos tipos de errores pueden provocar daños severos al sistema y esta ayuda permite al usuario actuar rápidamente sobre el error generado. Para ello, el sistema tendrá un ícono indicador, alarmas sonoras y visuales al momento de generarse este tipo de error.

El administrador puede configurar el sistema para recibir mails de los mensajes de error, en los casos cuya prioridad sean *critic* o *emerg*.

Idiomas del sistema: El sistema está implementado tanto para español como para inglés, dependiendo del idioma del sistema operativo.

3.1.3. Requerimientos no funcionales

Los requerimientos no funcionales especifican propiedades del sistema como restricciones de ambiente y desarrollo, rendimiento, dependencias de plataformas, mantenimiento y confiabilidad, etc. Se consideran las siguientes propiedades:

Mantenimiento y actualización

El sistema debe poseer la capacidad de permitir a los usuarios hacer mantenimiento de código y actualizaciones al software. Dado que el código es dado en el sistema y siendo este libre, este código puede ser tomado y modificado para generar mejoras y actualizaciones al sistema; para esto el código debe estar bien documentado, las variables empleadas deben ser entendibles y las funciones reutilizables.

Seguridad

Dado que es un administrador de logs, solo usuarios con permisos de root[8] podrán hacer cambios en la configuración del *syslog* y cambios en la configuración del sistema, los usuarios normales, es decir sin privilegios, no tendrán acceso por no contar con los permisos necesarios.

Usabilidad

El sistema debe proveer a los administradores la capacidad de configurar fácilmente el *syslog.conf* y de poder entender los reportes y las estadísticas sin muchos pasos y mucho esfuerzo. Debe proveer también una interfaz que sea fácil de aprender y sencilla de usar: el uso de tooltips[9] permite al usuario identificar más fácilmente una funcionalidad, además debe generar mensajes de error entendibles al usuario.

Disponibilidad

La disponibilidad del sistema debe ser contínua con un nivel de servicio para los usuarios de 7 días x 24 horas, garantizando un esquema adecuado que permita ante una posible falla de la solución en cualquiera de sus componentes, contar con una contingencia.

Rendimiento

El sistema debe presentar respuestas aceptables a las peticiones de los usuarios al momento de consultar los reportes o las estadísticas, ya que éstas son generadas desde

---

[8] En sistemas operativos del tipo Unix, root es el nombre convencional de la cuenta de usuario que posee todos los derechos en todos los modos (mono o multi usuario). root es también llamado superusuario.

[9] Mensajes descriptivos sobre una acción, son visibles al poner el puntero del mouse sobre el botón.

datos almacenados en base de datos, así mismo al momento de visualizar los logs, éstos deben poseer tiempos de respuesta muy pequeños.

También debe permitir la eliminación de logs repetidos de baja prioridad, pues es común incurrir en mensajes que suelen repetirse con una alta frecuencia generando miles de registros de logs. Esta característica debe solicitarse a voluntad del administrador, brindando así un mejor rendimiento de la base de datos.

Eficiencia

Debe advertir al administrador seguidamente generando el mensaje de error *critic* o *emerg*, por medio de ayudas visuales o con mails en caso de que el administrador no se encuentre en su equipo. Además debe llevar un control de cambios de los diversos mensajes de error al momento de pasar de una prioridad a otra.

Dado que los logs crecen en tamaño, el sistema debe ofrecer un medio de rotación de logs, esto consiste en crear respaldos hacia una fecha dada y comprimirlas, manteniendo así el historial de logs y no llenando espacio en disco

Operatividad

El sistema debe ser fácil de operar, y que el nivel de soporte a los usuario sea bajo.

Explicativo

Dado que son miles de logs generados, algunos de esos mensajes son poco entendibles para los usuarios, por lo cual la interfaz debe proveer descripciones sobre qué tipo de mensaje se generó, quién lo generó, cuándo se generó y su respectiva prioridad de error.

3.2 Análisis de tecnologías

Para esta tesis, vamos a clasificar las tecnologías en tres partes: versiones de syslog, familias Unix y algunos lenguajes comunes aplicables a Unix.

Versiones de Syslog

Existen algunas versiones en el mercado de syslog, como la tradicional Syslog, Metalog y SyslogNG (Next Generation). A continuación hablaremos sobre estas tecnologías.

- *Syslog [13][14]*

  Fue desarrollado por Eric Allman en 1980, como parte del control de correos en SENDMAIL, el servicio syslog tradicional se presentó en Septiembre de 1983 en la Universidad de California (Berkeley). Varios sistemas operativos implementaron este servicio, especialmente Unix y Linux.

  Consiste en que todos los mensajes generados por el sistema puedan ser controlados mediante un archivo de configuración *syslog.conf*, en el cual se detalla la funcionalidad del sistema que generó el evento y su prioridad.

- *Metalog [15]*

  Es un reemplazo moderno, eficaz y sencillo de *syslogd* y *klogd*. Permite registrar los mensajes en función de su *facility*, nivel, programa que lo genera e incluso con expresiones regulares de Perl. Los logs pueden ser rotados cuando estos alcanzan un cierto tamaño y es fácil de configurar al igual que el syslog.

- *Syslog-NG[16]*

  Es una versión mejorada del syslog creada por BALABIT. Permite generar filtros por expresiones regulares en los logs, no solo utiliza UDP para el envío de mensajes sino también TCP. Se creó mejorando las opciones del tradicional syslog, como encriptamiento, flexibilidad al guardar los logs, especialmente para redes centralizadas.

Familias de Unix

A continuación hablaremos de las familias de Unix en las cuales se ha probado este proyecto de tesis, dando previamente una breve introducción de Unix.

Unix [17]

Es un sistema operativo portable, multitarea y multiusuario; desarrollado en principio por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie y Douglas McIlroy.

Término familia en Unix

Técnicamente, Unix se refiere a una familia de sistemas operativos que comparten criterios de diseño e interoperabilidad en común. Esto no quiere decir que las diversas familias creadas durante 20 años compartan código o propiedad intelectual.

Distribuciones GNU/Linux

- *Ubuntu [18]*

  Basada en Debian GNU/Linux[10], Ubuntu es un sistema operativo de código abierto desarrollado en torno al kernel Linux. La filosofía Ubuntu se basa en los siguientes principios: que el software debe ser gratuito, que la gente debe poder usar el software en su lengua materna y debe poder hacerlo independientemente de cualquiera sean sus limitaciones; además, la gente debe ser libre de personalizar o modificar el software del modo que crea más conveniente.

[10] El proyecto GNU nació en 1984 de la iniciativa de Richard Stallman con el objetivo de desarrollar un sistema operativo basado en Unix, pero libre.

Sale una nueva versión al mercado cada 6 meses y es patrocinado por Canonical Ltd., una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth.

- *Fedora [19]*

  Fedora apareció en el 2003 como parte del anterior Red Hat Linux[11], muchos de los desarrolladores que creaban código para la versión Red Hat permitieron la creación de esta nueva distribución, posee interfaz amigable y su código es libre, una característica que posee, es que las modificaciones se la hace directamente sobre el código fuente, siendo esto útil para cualquiera de las versiones anteriores de Fedora.

Distribuciones BSD

- *FreeBSD [20]*

  El proyecto FreeBSD nació en los inicios de 1993 por una escisión parcial de los 3 coordinadores del "Unofficial 386BSD Patchkit": Nate William, Rod Grimes y Jordan K. Hubbard, basándose en la versión 386BSD de Bill Jolitz.

  Es un avanzado sistema operativo para arquitecturas x86 compatibles, amd64 compatibles, UltraSPARC, IA-64, PC-98 y ARM. FreeBSD es un derivado de BSD, la versión de Unix desarrollada en la Universidad de California, Berkeley.

- *MAC OS X Tiger [21]*

  Mac OS, abreviatura de Macintosh Operating System (Sistema Operativo de Macintosh), es el nombre del primer sistema operativo de Apple para los ordenadores Macintosh. El Mac OS original fue el primer sistema operativo con una interfaz gráfica de usuario en tener éxito. Mac OS X es un sistema operativo basado en Unix, pero donde el gestor de

---

[11] Es una distribución Linux creada por Red Hat, que fue una de las más populares en los entornos de usuarios domésticos.

ventanas X11, característico de estos sistemas, ha sido sustituido por otro denominado Aqua, desarrollado íntegramente por Apple.

Lenguajes de Programación

A continuación se mencionarán algunos lenguajes de programación propios en Unix.

- *C/C++[22]*

  Es un lenguaje de programación creado en 1972 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, es un lenguaje orientado a la implementación de sistemas operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

  Se trata de un lenguaje débilmente diseñado para medio nivel pero con características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

- *Perl [23]*

  Perl está basado en lenguajes como C, sh, awk y sed (lenguajes propios de sistemas Unix), pero su enfoque es más práctico y fácil que los mencionados anteriormente. Es por ello que un programador que haya trabajado con el lenguaje C y los otros tendrá menos problemas en entenderlo y utilizarlo rápidamente.

  Perl no limita el tamaño de los datos con los que trabaja, el límite lo pone la memoria que en ese momento se encuentre disponible.

- *AWK [24]*

Llamado así por los apellidos de sus inventores Aho, Weinberger y Kernighan. Lenguaje de programación antiguo diseñado para el procesamiento de datos basados en texto. Cuando es escrito en minúsculas, *awk* hace referencia al programa de Unix que corría otros programas escritos en este lenguaje de programación. El AWK es un lenguaje de programación que usa los tipos de dato cadena, arreglos indexados tipo string y expresiones regulares.

- *Python [25]*

  El creador del lenguaje es un europeo llamado Guido Van Rossum. Ayudado y motivado por su experiencia en la creación de otro lenguaje llamado ABC. El objetivo de Guido era cubrir la necesidad de un lenguaje orientado a objetos de sencillo uso que sirviese para tratar diversas tareas dentro de la programación que habitualmente se hacía en Unix usando C.

  Python es un lenguaje de interpretado independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas Web. Al ser lenguaje interpretado, no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.

3.3 Descripción de casos de uso y escenarios

Los casos de uso se refieren a qué realiza el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

El sistema administrador del Syslog de Unix se encuentra definido mediante los siguientes casos de uso, ilustrados por medio de la Figura 2.

1. Usuario ingresa al sistema.
2. Usuario escoge Configuración Original.

3. Usuario escoge Configuración Personalizada.
4. Usuario modifica Configuración Personalizada.
5. Usuario visualiza archivos de logs
6. Usuario realiza filtros sobre los archivos de logs.
7. Usuario limpia filtros en visualizador de logs
8. Usuario genera reportes por filtros.
9. Usuario exporta reporte como XML.
10. Usuario genera estadísticas.
11. Usuario genera alertas.
12. Usuario bloquea un tipo de mensaje de log.
13. Usuario desbloquea un tipo de mensaje de la lista de bloqueados
14. Usuario genera respaldos de los logs.
15. Usuario restaura logs.
16. Usuario elimina logs.

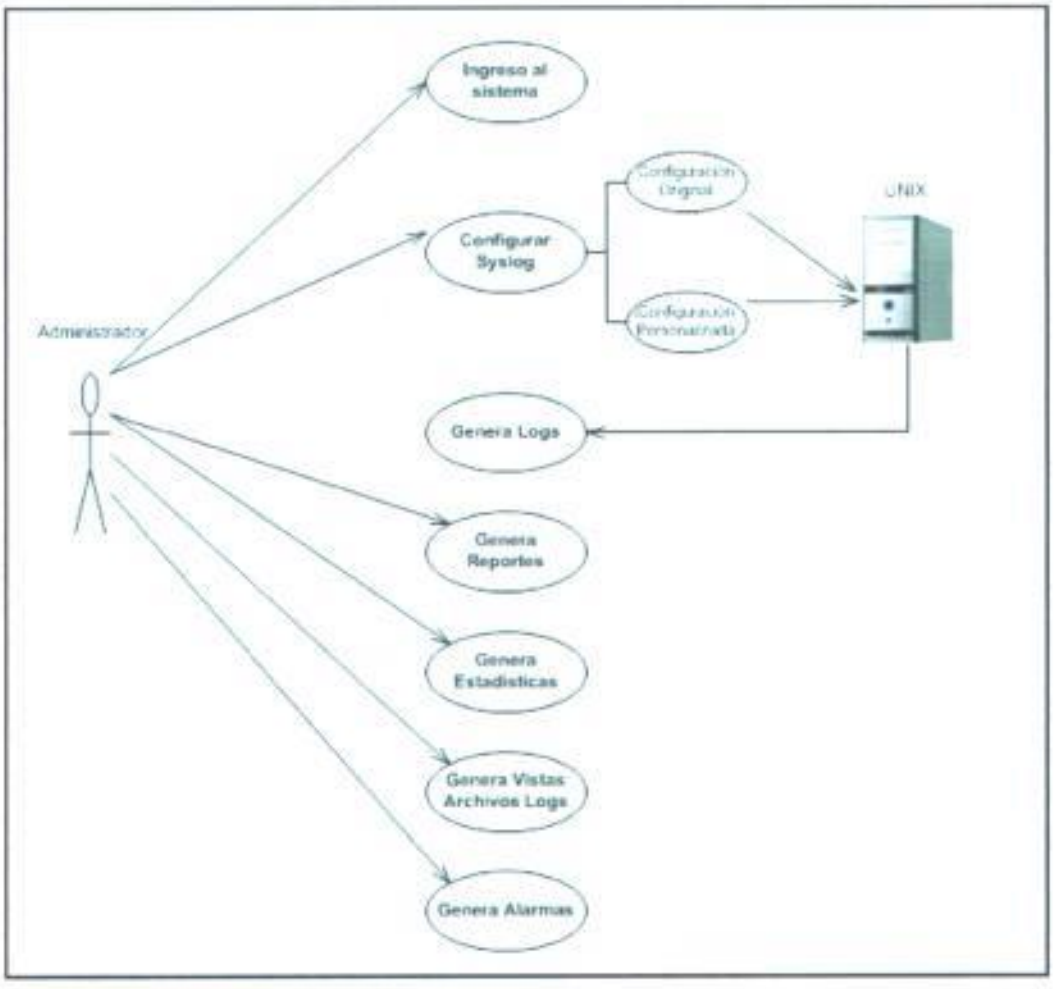

Figura 2. Diagrama de casos de uso.

La definición de cada caso de uso se describe de la siguiente manera:

Nombre: El nombre del caso de uso.

Descripción: Breve reseña acerca del caso de uso.

Precondiciones: Son los hechos que se han de cumplir para que el flujo de eventos se pueda llevar a cabo.

Flujo Normal: Es la ejecución normal y exitosa del caso de uso.

Flujo Alterno: Son los que nos permiten indicar qué es lo que hace el sistema en los casos menos frecuentes e inesperados.

Excepciones: Casos en los cuales no se cumple la precondición.

Pos Condiciones: Son los hechos que se han de cumplir si el flujo de eventos normal se ha ejecutado correctamente.

1. Usuario Ingresa al sistema

Nombre:	Usuario ingresa al sistema.
Descripción:	
Usuario ingresa al sistema de administración de logs para Unix.	
Actores:	
Usuario del sistema.	
Precondiciones:	
• Usuario tiene cuenta en el sistema operativo con permisos asignados. • Servicio *Syslogd* debe estar habilitado.	
Flujo Normal:	
a) Usuario tiene permisos de Root. b) Usuario ingresa al sistema operativo de la distribución dada. c) Usuario accede al sistema de logs. d) Sistema no valida acceso, dado que es administrador del sistema por tener permisos de Root.	
Flujo Alterno:	
a) Usuario no tiene permisos de Root. b) Usuario ingresa al sistema operativo de la distribución dada. c) Usuario accede al sistema de logs. d) El sistema deniega el acceso por no tener los permisos necesarios dentro del sistema operativo para visualizar ni administrar esta información	
Excepciones:	

Pos condiciones:
• Usuario ingresa al sistema de logs. • Se registra fecha de ingreso al sistema. • Por cada ingreso se mostrará la fecha de último ingreso al sistema.

Figura 3. Caso de uso 1: Usuario ingresa al sistema.

2. Usuario escoge Configuración Original.

Nombre:	Usuario Personaliza Configuración Original.
Descripción: Usuario escoge la configuración original para configurar el archivo *syslog.conf* del sistema operativo.	
Actores: Usuario del sistema.	
Precondiciones: Usuario ha ingresado al sistema de administración de logs.	
Flujo Normal: a) Usuario escoge Configuración del Syslog b) Usuario escoge Configuración Original. c) Usuario da clic en "*Finalizar*".	
Excepciones: Usuario cancela las opciones de configuración.	
Pos condiciones: • La configuración indicada en el archivo *syslog.conf* del S.O. se emplea como base del sistema administrador de logs, que servirá para la generación de reportes, estadísticas y demás utilidades proporcionadas por el sistema.	

Figura 4. Caso de uso 2: Usuario escoge Configuración Original.

3. **Usuario escoge Configuración Personalizada.**

Nombre:	Usuario escoge Configuración Personalizada.
Descripción:	Usuario escoge personalizar la configuración del archivo *syslog.conf* por medio de elección simple que consiste en escoger el servicio y sus niveles de error, los cuales serán mostrados al usuario en el visor de logs.
Actores:	Usuario del sistema.
Precondiciones:	Usuario ha ingresado al sistema de administración de logs.
Escenarios:	3.1. Usuario escoge selectores a operar y da clic en "*Finalizar*". 3.2. Usuario no escoge ningún selector y da clic en "*Finalizar*".
Excepciones:	Usuario cancela las opciones de configuración.
Pos condiciones:	• La configuración del *syslog.conf* se guarda a las preferencias del usuario.

Figura 5. Caso de uso 3: Usuario escoge Configuración Personalizada.

Escenario 3.1 Usuario escoge selectores a operar y da clic en "*Finalizar*".

Suposiciones:

- ✓ Usuario selecciona selector (servicio y prioridad).
- ✓ Usuario agrega selector a la Lista de Configuraciones.
- ✓ Usuario da clic en "*Finalizar*".

Resultados:

✓ Los cambios son aceptados y el *syslog.conf* se actualiza.

Figura 6. Caso de uso 3: Usuario escoge Configuración Personalizada - Escenario 3.1.

Escenario 3.2 Usuario no escoge ningún selector y da clic en "*Finalizar*".
Suposiciones:
✓ Usuario no selecciona ningún selector.
✓ Usuario da clic en "*Finalizar*".
Resultados:
✓ Un mensaje de error es generado.

Figura 7. Caso de uso 3: Usuario escoge Configuración Personalizada - Escenario 3.2.

4. Usuario modifica Configuración Personalizada.

Nombre:	Usuario modifica Configuración Personalizada.
Descripción:	Ya teniendo una configuración personalizada, el usuario escoge personalizar la configuración del archivo *syslog.conf* por medio de la elección simple de que desea que el sistema le notifique (correo, núcleo, etc.).
Actores:	Usuario del sistema.
Precondiciones:	Usuario ha ingresado al sistema de administración de logs.
Escenarios:	4.1. Usuario modifica selectores a operar y da clic en "*Finalizar*". 4.2. Usuario no modifica ningún operador y da clic en "*Finalizar*".
Excepciones:	Usuario cancela las opciones de configuración.

Pos condiciones: • La configuración del *syslog.conf* se actualiza a las preferencias del usuario.

Figura 8. Caso de uso 4: Usuario modifica Configuración Personalizada.

Escenario 4.1 Usuario modifica selectores a operar y da clic en "*Finalizar*". **Suposiciones:** ✓ Usuario modifica selectores utilizados por los que va a utilizar. ✓ Usuario da clic en "*Finalizar*". **Resultados:** ✓ Los cambios son aceptados y el *syslog.conf* se actualiza.

Figura 9. Caso de uso 4: Usuario modifica Configuración Personalizada - Escenario 4.1

Escenario 4.2 Usuario no modifica ningún selector y da clic en "*Finalizar*". **Suposiciones:** ✓ Usuario no modifica ningún selector. ✓ Usuario da clic en "*Finalizar*". **Resultados:** ✓ Los cambios son aceptados y el *syslog.conf* se actualiza.

Figura 10. Caso de uso 4: Usuario modifica Configuración Personalizada - Escenario 4.2.

5. Usuario Visualiza archivos de logs.

Nombre:	Usuario visualiza archivos de logs.
Descripción: Usuario selecciona vista de logs, dando clic en cualquiera de los logs generados, será presentado en formato de columnas al usuario, tanto por fecha, hora, máquina, demonio y mensaje.	

Actores: Usuario del sistema.
Precondiciones: ▪ Usuario ha ingresado al sistema de administración de logs. ▪ Usuario ya ha configurado el *syslog.conf*.
Flujo Normal: Usuario da clic en el log que desea visualizar.
Excepciones:
Pos condiciones: ▪ Dependiendo del archivo log seleccionado, este será mostrado al lado derecho de la lista de logs.

Figura 11. Caso de uso 5. Usuario visualiza archivos de logs.

6. Usuario realiza filtros sobre los archivos de logs.

Nombre:	Usuario realiza filtros sobre los archivos de logs.
Descripción: Usuario selecciona una vista de log, en la barra de filtros escoge que desea visualizar por un campo o por el conjunto de campos (fecha, hora, máquina, demonio y mensaje).	
Actores: Usuario del sistema.	
Precondiciones: • Usuario ha ingresado al sistema de administración de logs. • Usuario ya ha configurado el *syslog.conf* • Usuario selecciona vista de logs. • Usuario escoge algún filtro de la barra de filtros.	

Escenarios:

6.1. Usuario filtra por fecha.

6.2. Usuario filtra por demonio.

6.3. Usuario filtra por mensaje.

Excepciones:

Pos condiciones:

- Dependiendo de los filtros aplicados, se visualizara en el visor de logs.

Figura 12. Caso de uso 6: Usuario realiza filtros sobre los archivos de logs.

Escenario 6.1 Usuario filtra por fecha.

Suposiciones:

- ✓ Usuario escoge vista de log.
- ✓ Usuario ingresa fecha inicio y fecha fin para el filtro.

Resultados:

- ✓ En el visor del log se visualiza lo indicado por el usuario.

Figura 13. Caso de uso 6: Usuario realiza filtros sobre los archivos de logs. - Escenario 6.1.

Escenario 6.2 Usuario filtra por demonio.

Suposiciones:

- ✓ Usuario escoge vista de log.
- ✓ Usuario ingresa demonio a filtrar.

Resultados:

- ✓ En el visor del log se visualiza lo indicado por el usuario.

Figura 14. Caso de uso 6: Usuario realiza filtros sobre los archivos de logs. - Escenario 6.3.

Escenario 6.3 Usuario filtra por mensaje.

Suposiciones: ✓ Usuario escoge vista de log. ✓ Usuario ingresa todo o parte del mensaje. **Resultados:** ✓ En el visor del log se visualiza lo indicado por el usuario.

Figura 15. Caso de uso 6: Usuario realiza filtros sobre los archivos de logs. - Escenario 6.4.

7. Usuario limpia filtros en archivos logs.

Nombre:	Usuario limpia filtros en archivos logs.
Descripción: Después de hacer una búsqueda por los archivos logs, el usuario limpia el resultado de las búsquedas.	
Actores: Usuario del sistema.	
Precondiciones: • Usuario ha ingresado al sistema de administración de logs. • Usuario ya ha configurado el *syslog.conf* • Usuario genero un filtro sobre un log previamente.	
Flujo Normal: a) Se realizo una búsqueda sobre el log. b) Se da clic en "*Restaurar*".	
Excepciones: Si no había búsqueda previa, no pasa nada.	
Pos condiciones: • La vista del archivo log es restaurada a su estado original.	

Figura 16. Caso de uso 7: Usuario limpia filtros en archivos logs.

8. **Usuario genera Reportes por filtros.**

Nombre:	Usuario genera reportes por filtros.
Descripción: Permite generar un reporte a partir de filtros hechos a los campos del registro de eventos (fecha, demonio, servicio, nivel y mensaje).	
Actores: Usuario del sistema.	
Precondiciones: • Usuario ha ingresado al sistema de administración de logs. • Usuario ya ha configurado el *syslog.conf*.	
Flujo Normal: a) Se escoge Reporte por filtros. b) Usuario escoge filtros a su criterio.	
Excepciones: El filtro no generó coincidencia alguna.	
Pos condiciones: • Dependiendo del filtro escogido, se visualiza el reporte.	

Figura 17. Caso de uso 8: Usuario genera Reportes por filtros.

9. **Usuario exporta reportes como XML.**

Nombre:	Usuario exporta reportes como XML.
Descripción: Permite exportar un reporte como XML.	
Actores: Usuario del sistema.	

Precondiciones: • Usuario ha ingresado al sistema de administración de logs. • Usuario ya ha configurado el *syslog.conf* • Usuario genero un reporte.
Flujo Normal: a) Se genera el reporte. b) Usuario da clic en "*Exportar*".
Excepciones:
Pos condiciones: • El reporte es exportado como XML

Figura 18. Caso de uso 10: Usuario exporta reportes como XML.

10. Usuario genera Estadisticas.

Nombre:	Usuario genera estadísticas.
Descripción: Permite generar estadísticas basadas en la selección del usuario.	
Actores: Usuario del sistema.	
Precondiciones: • Usuario ha ingresado al sistema de administración de logs. • Usuario ya ha configurado el *syslog.conf*. • Usuario escoge un tipo de estadísticas.	

Escenarios: 10.1. Estadísticas de servicio por incidencias. 10.2. Estadísticas de prioridad por incidencias. 10.3. Estadísticas de servicio por fecha. 10.4. Estadísticas de prioridad por fecha.
Excepciones: Parámetros no generaron ninguna gráfico estadístico.
Pos condiciones: ▪ Si los parámetros generan datos, se genera los gráficos estadísticos.

Figura 19. Caso de uso 11: Usuario genera Estadísticas.

Escenario 10.1 Estadísticas de servicio por incidencias. **Suposiciones:** ✓ Usuario escoge Estadísticas de servicio por incidencias. ✓ Usuario escoge filtro tanto por servicio como por fecha. **Resultados:** ✓ Se mostrara un gráfico estadístico de acuerdo a los filtros hechos por el usuario.

Figura 20. Caso de uso 10: Usuario genera Estadísticas. - Escenario 10.1.

Escenario 10.2 Estadísticas de prioridad por incidencias. **Suposiciones:** ✓ Usuario escoge Estadísticas de prioridad por incidencias. ✓ Usuario escoge filtro tanto por nivel como por fecha. **Resultados:** ✓ Se muestra un gráfico estadístico de acuerdo a los filtros hechos por el usuario.

Figura 21. Caso de uso 10: Usuario genera Estadísticas. - Escenario 10.2.

Escenario 10.3 Estadísticas de servicio por fecha.
Suposiciones: ✓ Usuario escoge Estadísticas de servicio por fecha. ✓ Usuario escoge filtro tanto por servicio como por fecha.
Resultados: ✓ Se muestra un gráfico estadístico de acuerdo a los filtros hechos por el usuario.

Figura 22. Caso de uso 10: Usuario genera Estadísticas. - Escenario 10.3.

Escenario 10.4 Estadísticas de prioridad por fecha.
Suposiciones: ✓ Usuario escoge Estadísticas de prioridad por fecha. ✓ Usuario escoge filtro tanto por prioridad como por fecha.
Resultados: ✓ Se muestra un gráfico estadístico de acuerdo a los filtros hechos por el usuario.

Figura 23. Caso de uso 10: Usuario genera Estadísticas. - Escenario 10.4.

11. Usuario genera alertas.

Nombre:	Usuario genera alertas.
Descripción: El usuario genera alertas basándose en el servicio y el nivel de error de un mensaje de log. Además, indica dónde desea que se le notifique al administrador (correo, teléfono o notificación al usuario).	
Actores: Usuario del sistema.	
Precondiciones: Usuario ha ingresado al sistema de administración de logs.	

Flujo Normal: a) Usuario ingresa al sistema. b) Usuario escoge agregar nueva alerta. c) Basándose en lo ingresado, la alerta es mostrada al administrador
Excepciones:
Pos condiciones: ▪ Dependiendo de los datos ingresados, la alerta será mostrada al administrador.

Figura 24. Caso de uso 12: Usuario genera alertas.

12. Usuario bloquea un tipo de mensaje de log

Nombre:	Usuario bloquea un tipo de mensaje de log.
Descripción: Usuario bloquea aquellos mensajes que no desea sean mostrados ni usados para generar las estadísticas y reportes.	
Actores: Usuario del sistema.	
Precondiciones: • Usuario ha ingresado al sistema de administración de logs. • Usuario ha seleccionado una vista de logs.	
Flujo Normal: a) Usuario accede a vista de logs. b) Usuario selecciona un mensaje de log. c) Usuario da clic derecho y selecciona "*Enviar a lista de exclusión*".	
Excepciones: Usuario cancela bloqueo de mensajes en Lista de Exclusión.	

Pos condiciones:
• El Mensaje se ha ingresado a la Lista de Exclusión.

Figura 25. Caso de uso 13: Usuario bloquea un tipo de mensaje de log.

13. Usuario desbloquea un tipo de mensaje de log de la Lista de Exclusión.

Nombre:	Usuario desbloquea un tipo de mensaje de log de la Lista de Exclusión.
Descripción:	
Usuario desbloquea mensajes de la Lista de Exclusión.	
Actores:	
Usuario del sistema.	
Precondiciones:	
• Usuario ha ingresado al sistema de administración de logs. • Usuario ha seleccionado Lista de Exclusión.	
Flujo Normal:	
a) Usuario ingreso a Lista de Exclusión. b) Usuario selecciona un mensaje de la lista de exclusión. c) Usuario da clic derecho y selecciona "*Quitar*".	
Excepciones:	
Usuario cancela desbloqueo de mensajes en Lista de Exclusión.	
Pos condiciones:	
• El Mensaje es desbloqueado de la Lista de Exclusión.	

Figura 26. Caso de uso 14: Usuario desbloquea un tipo de mensaje de log de la Lista de Exclusión.

14. Usuario genera respaldos de los logs.

Nombre:	Usuario genera respaldos de los logs.

Descripción: Usuario genera respaldos de los logs por filtros como la fecha, servicio y nivel.
Actores: Usuario del sistema.
Precondiciones: • Usuario ha ingresado al sistema de administración de logs. • Usuario ha seleccionado Respaldo de logs.
Flujo Normal: a) Usuario ingresa a Respaldo de logs. b) Usuario escoge un filtro para el respaldo. c) Usuario da clic en "*Respaldar*".
Excepciones: Usuario cancela Respaldo de logs.
Pos condiciones: • Se genera el respaldo con los filtros escogidos por el usuario.

Figura 27. Caso de uso 14: Usuario genera respaldo de los logs.

15. Usuario restaura logs.

Nombre:	Usuario restaura logs.
Descripción: Usuario restaura logs a la base de datos.	
Actores: Usuario del sistema.	
Precondiciones: • Usuario ha ingresado al sistema de administración de logs. • Usuario ha seleccionado Restaurar logs.	

Flujo Normal: a) Usuario ingresa a Restaurar logs. b) Usuario escoge el filtro para restaurar los logs. c) Usuario da clic en "*Restaurar*".
Excepciones: Usuario cancela Restaurar logs.
Pos condiciones: • Los logs que se encontraban en el archivo se cargan a la base de datos de logs.

Figura 28. Caso de uso 15: Usuario restaura logs.

16. Usuario elimina logs por filtros.

Nombre:	Usuario elimina logs por filtros.
Descripción: Basándose en filtros, el usuario elimina logs.	
Actores: Usuario del sistema.	
Precondiciones: • Usuario ha ingresado al sistema de administración de logs. • Usuario ha seleccionado Eliminar logs.	
Flujo Normal: a) Usuario ingresa a Eliminar logs. b) Usuario escoge un filtro para la eliminación de logs. c) Usuario da clic en eliminar.	

Excepciones: Usuario cancela Eliminar logs.
Pos condiciones: • Se eliminan los logs de la base de datos.

Figura 29. Caso de uso 16: Usuario elimina logs por filtros.

3.4 Análisis de la interacción Hombre – Máquina

La interacción Hombre-Máquina estudia para todos los sistemas computacionales, la forma en que estos deben ser diseñados, evaluados e implementados con el fin de que sean beneficiosos para el uso humano. Se basa en el concepto de que un sistema debe ser útil y que permita lograr objetivos con efectividad y eficiencia y que los resultados puedan ser los deseados. Para esto el sistema debe cumplir con las siguientes normas:

- **Capacidad de aprendizaje:** El tiempo y esfuerzo requerido para que un usuario experto o no experto alcance un determinado nivel de ejecución en un sistema dado. Para esto los nombres de las acciones deben ser claros y coherentes con la acción, a fin de que los usuarios sepan instintivamente la funcionalidad de los distintos elementos de entrada.
- **Rendimiento:** La velocidad en la ejecución de las tareas y el número y tipo de errores cometidos por el usuario en su realización.
- **Satisfacción:** Medidas del confort, la aceptabilidad y la actitud positiva generada por el sistema en las personas afectadas por su uso.
- **Flexibilidad:** La capacidad del sistema de poder trabajar con diferentes métodos en función del nivel de experiencia del usuario. Para nuestro caso, la configuración del *syslog.conf* no depende de la experiencia ni del conocimiento del usuario, no se emplean términos técnicos avanzados, los reportes y estadísticas también son de fácil entendimiento para cualquier tipo de usuario.

- **Efectividad:** El grado de exactitud con que el sistema completa la/s tarea/s para las que está diseñado.
- **Eficiencia:** Hace referencia al número de pasos que el usuario debe llevar a cabo para completar la tarea.
- **Visibilidad:** Las opciones o funcionalidades que brinde el sistema deben estar en forma clara y entendible; en el caso de botones, los íconos usados deben ser coherentes con la acción a realizar, el uso de tooltips es aplicado.
- **Manejo de Errores:** El sistema debe proveer mensajes de error claros y fáciles de entender, evitando usar términos técnicos y sobretodo que el sistema permita recuperarse de un error cuando este ocurra.
- **Uso de colores:** Debe evitarse el uso de muchos colores y combinaciones con colores fuertes.

3.5 Herramientas empleadas en el desarrollo del sistema

El sistema operativo sobre el cual se basa el proyecto de tesis es Unix. Las pruebas del sistema se las realizará sobre familias de Unix como GNU/Linux y BSD. Existen muchas distribuciones de estas dos familias, de las cuales se ha elegido por el lado de GNU/Linux a Fedora (basado en Red Hat) y Ubuntu (basado en Debian); y por la familia de BSD a FreeBSD y Mac OS X las cuales usan las plataformas elaboradas por BSD.

Para la prueba de estos sistemas operativos, se utilizaron máquinas virtuales[12] donde se encontrará el software instalado y probado en cada uno de estos ambientes. Para ello se empleará la herramienta VMWARE WORKSTATION a fin de crear las 4 máquinas virtuales de los sistemas operativos mencionados anteriormente.

[12] Software que permite simular un sistema operativo, se almacenan como archivos.

De las bases de datos disponibles en el mercado se usó SQLITE, que es una base de datos relacional desarrollada en C. Esta base tiene la particularidad de que su motor no es un proceso independiente con el cual el programa principal se comunica, sino que es parte integral del mismo (a diferencia de las bases de datos cliente-servidor), y utiliza llamadas simples de subrutinas y funciones reduciendo así la latencia de acceso a la base, reemplazando la comunicación entre procesos.

El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), es guardado como un sólo archivo estándar en la máquina host; lo cual se logra bloqueando todo el archivo de base de datos al principio de cada transacción.
Adicionalmente se hizo uso de la herramienta SQLITEBROWSER, empleado como visor de la base de datos.

El lenguaje utilizado para el desarrollo es Python, dado que posee características como comprobación de errores (mejores que en C) y al ser un lenguaje de muy alto nivel, tiene incluidos tipos de datos de alto nivel, como matrices flexibles y diccionarios, que llevarían días de programación en C. Dados sus tipos de datos más generales, se puede aplicar a un rango de problemas más amplio que *awk* o incluso Perl.

Python es un lenguaje interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. Los programas hechos en Python son más cortos dado que usa la indentación en lugar de Begin/End o llaves.

Para creación de la interfaz, se empleo el uso de GLADE, esta herramienta permite construir interfaces graficas para GTK+ y aplicaciones GNOME

Como herramienta de desarrollo se empleó KOMODO. Esta herramienta permite hacer compilaciones, debug, watches y breakpoints. Posee también característica para autocompletar si una función ha sido invocada.

CAPÍTULO 4

DISEÑO DEL SISTEMA

4.1 Diseño de la arquitectura del sistema

El sistema está basado en dos capas, la capa de generación de logs y la capa de extracción y manipulación de logs.

Capa de generación de logs

Esta capa muestra el proceso en que los sistemas Unix generan los mensajes de logs.

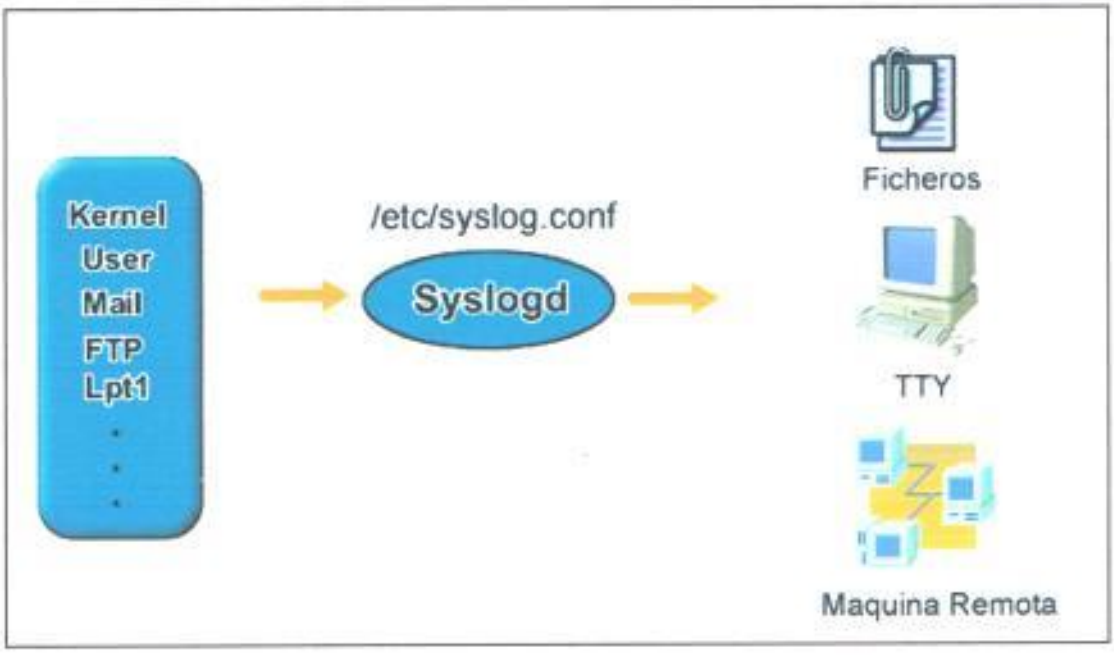

Figura 30. Arquitectura del *syslogd*.

Syslogd se carga automáticamente al momento de iniciar el sistema y basado en la información registrada en el archivo *syslog.conf* genera informes sobre el funcionamiento del sistema y lo almacena en las rutas especificadas en el archivo de configuración. Puede

recibir mensajes de diversos servicios como kernel u otros programas y puede almacenar información en destinos como máquinas remotas, archivos FIFO o en la maquina local, etc. (Figura 30).

Las líneas del *syslog.conf* que comienzan por '#' son comentarios, que son ignoradas de la misma forma que las líneas en blanco; si ocurriera un error al interpretar una de las líneas del archivo, se ignoraría la línea completa.

Capa de extracción y manipulación de logs

Esta capa toma los datos generados por el *syslogd* y los almacena en una base de datos SQLITE. Debido a que el sistema está desarrollado en Python, se necesita una librería que permita enlazar una base de datos SQLITE con Python, con este propósito se emplea Pysqlite. *Pysqlite* es una interface python para la base de datos relacional SQLITE.

Para la generación de ventanas y ambientes gráficos, se utiliza *GLADE3,* que es una herramienta de desarrollo visual adaptable a lenguajes de programación basados en Unix. GLADE no genera código fuente sino un archivo XML.

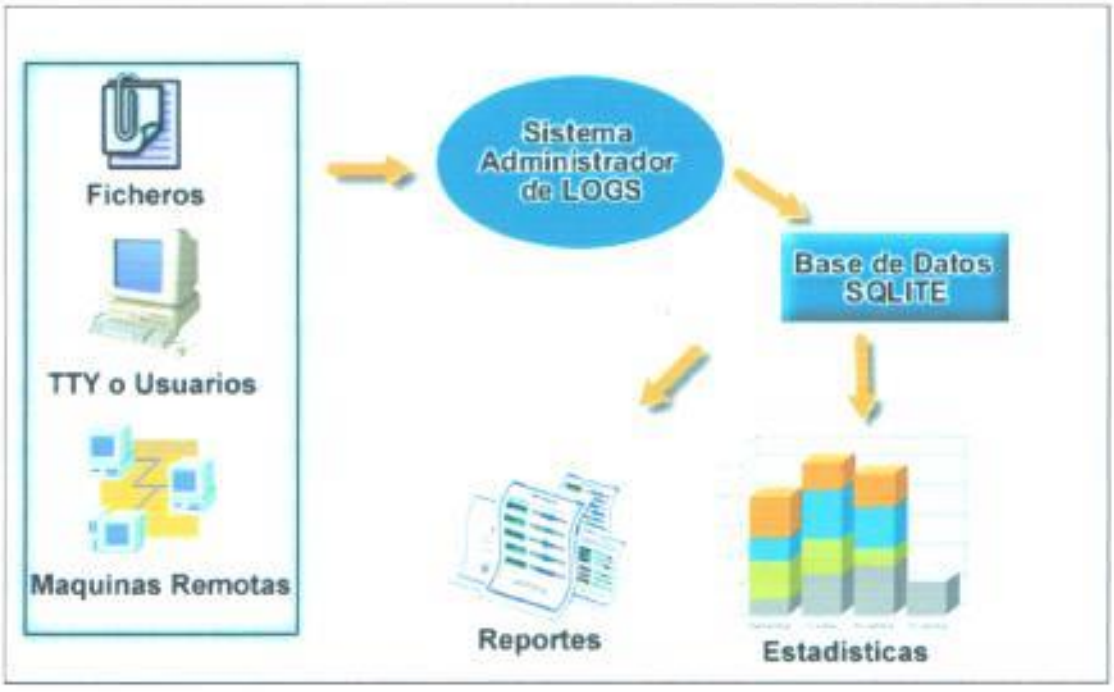

Figura 31. Arquitectura del *SyslogManager*.

Todas estas librerías junto con Python y SQLITE, son fácilmente descargables en cada uno de los sistemas Unix escogidos para este tema de tesis. En el caso de Fedora, se usa el comando YUM; para FREEBSD, las librerías y el software requerido están disponibles en su página oficial [20]; y para UBUNTU, se utiliza el comando APT-GET.

4.2 Diseño de la base de datos

4.2.1. Justificación del uso de la base de datos

Una base de datos es empleada en el sistema a fin de poder administrar los logs recibidos por el *syslogd*, permitiendo así realizar filtros, generar reportes y estadísticas de los archivos logs.

Dado que los mensajes de logs no se guardarán en un archivo de texto sino directamente en la base de datos, hace que los datos sean más seguros, evitando así que intrusos quienes originalmente podrían hurtar información de los archivos logs, puedan lograrlo debido a que se encuentran dentro de la base de datos.

Un aspecto importante al escoger una base de datos, es que sea multiplataforma, es decir que pueda ser usada en cualquier tipo de aplicación, sea ésta para Windows o para Linux y accesible a cualquier lenguaje de programación, en este caso Python.

Otra ventaja es su licencia abierta. SQLITE puede ser utilizado, modificado y distribuido y toda su documentación está disponible gratuitamente en Internet y existente para cualquier tipo de Unix.

SQLITE además no es pesado y es fácilmente manipulable, posee todas las sentencias de bases de datos y su motor de bases de datos en velocidad es similar al provisto por POSTGRESQL[13] y MYSQL[14] que son también motores de código abierto.

---

[13] Base de datos de software libre, relacional y orientada a objetos.

[14] Es un sistema de gestión de base de datos relacional, multi-hilo y multi-usuario

4.2.2. Diagrama entidad-relación

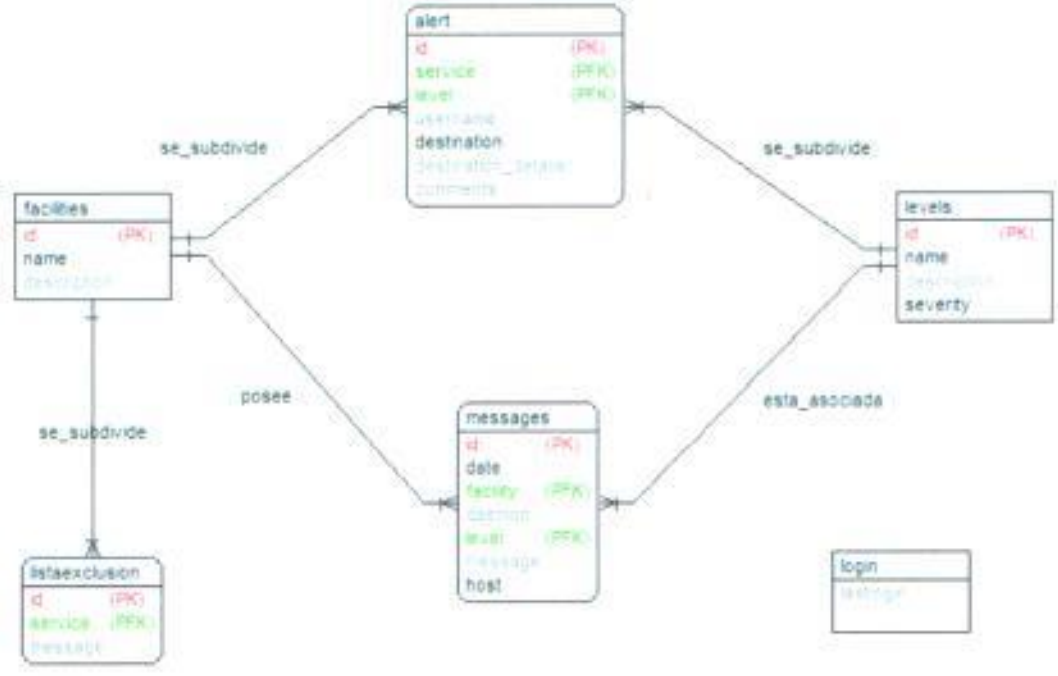

Figura 32. Modelo lógico del sistema.

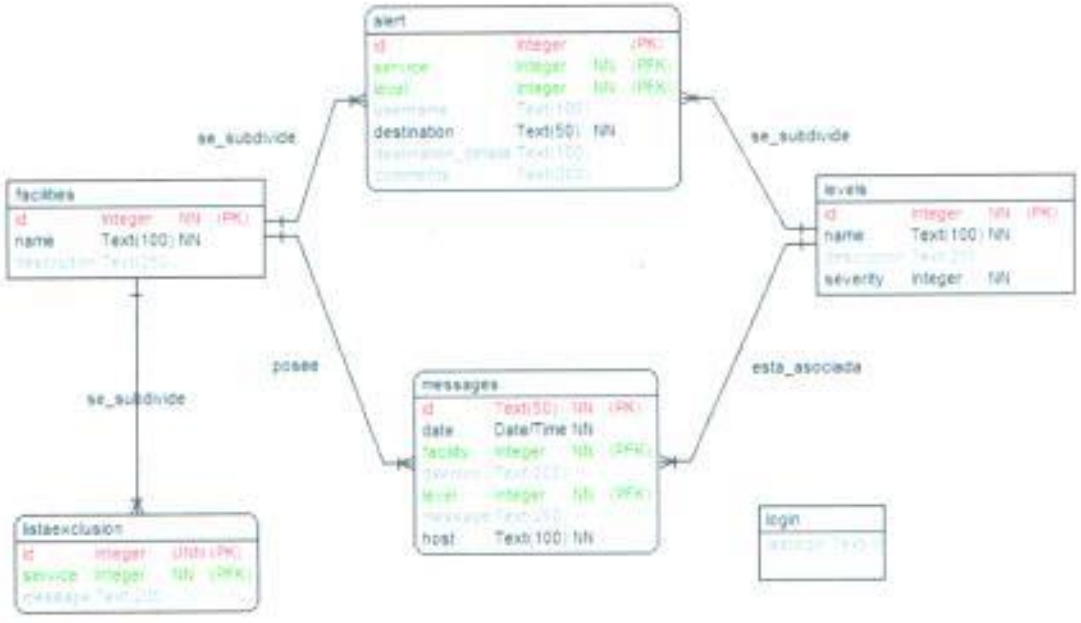

Figura 33. Modelo físico del sistema.

4.2.3. Descripción de las entidades

Referirse al **Anexo B**.

4.3 Diseño de interfaces con el usuario

En esta sección se definirán los flujos de ventanas con las opciones disponibles para el administrador del sistema.

4.3.1. Flujo de ventanas y layouts

El flujo de ventanas describe la secuencia de las ventanas que el administrador visualizará, además de la funcionalidad requerida para el sistema. Permite a su vez establecer la interfaz de usuario y se describen en términos de las tareas que se esperan que sean realizadas por estos, sin necesidad de detallar todos los casos excepcionales, como el control de errores.

Para este proyecto de tesis, el administrador podrá acceder a cuatro ventanas principales, indicadas en la Figura 34.

Figura 34. Flujo de ventanas general del sistema.

Configuración: Permite al usuario hacer modificaciones sobre la configuración del sistema, tanto para las alarmas como para lo que se administrara.

Estadísticas: Genera estadísticas por aparición y por fecha en gráfico circular, histogramas y líneas de tiempo.

Reportes: Genera reportes sobre los logs.

Vista de Logs: Visualiza los logs, permite hacer filtros sobre los mensajes generados.

La Figura 35 muestra el flujo de ventanas que obtiene el usuario al acceder a la sección de configuración.

Figura 35. Flujo de ventanas de la sección Configuración.

Agregar nueva alerta: Permite al administrador crear una nueva alerta la cual será notificada cuando esta ocurra. Será enviada al correo o a celular.

Lista alertas actuales: Muestra todas las alertas creadas por el administrador.

Lista de exclusión: Permite excluir un mensaje de log para que no sea visto por el usuario.

Configurar Syslog: Permite configurar el *syslog.conf* con las opciones de configuración original y configuración Personalizada.

Figura 36. Flujo de ventanas de sección Estadísticas.

Servicio por Incidencia: Mediante filtros por fecha y por servicio, permite mostrar las incidencias de los niveles en formato de gráfico circular e histogramas.

Prioridad por Incidencia: Mediante filtros por fecha y por nivel, permite mostrar las incidencias de los servicios en formato de gráfico circular y de histogramas.

Servicio por Fecha: Mediante filtros por fecha y por servicio, permite mostrar por días el número de apariciones de los niveles, en formato de líneas de tiempo por aparición.

Prioridad por Fecha: Mediante filtros por fecha y por nivel, permite mostrar por días el número de apariciones de los servicios, en formato de líneas de tiempo por aparición.

Figura 37. Flujo de ventanas de sección

Reportes por filtros: Mediante filtros como fecha, servicio, nivel y búsqueda por palabras del mensaje.

Exportar: Permite exportar el resultado del filtro en formato XML.

Figura 38. Flujo ventanas sección vistas de logs

Búsquedas: Permite realizar búsquedas sobre los archivos de logs.

Figura 39. Flujo ventanas sección Mantenimiento

Respaldo de logs: Mediante filtros por fecha, servicio y nivel, permite al administrador crear respaldos de los logs.

Restaurar logs: Permite restaurar logs que han sido previamente respaldados.

Eliminar logs: Mediante filtros por fecha, servicio y nivel, permite al administrador eliminar logs.

4.3.2. Diseño de la interacción del usuario

El diseño de la interacción con el usuario para la demostración de este proyecto de tesis se la realizó basándose en las indicaciones dadas en el análisis de interacción Hombre – Máquina.

Basándose en los conceptos de usabilidad de un sistema [26] el diseño de interacción se lo basa en los siguientes conceptos:

- ✓ **Estructura:** organiza con significado.
- ✓ **Simplicidad:** haz fáciles las tareas comunes.
- ✓ **Visibilidad:** muestra toda aquella información necesaria para una tarea.
- ✓ **Retroalimentación:** mantén informados a los usuarios.
- ✓ **Tolerancia:** permite cancelar, deshacer, volver.
- ✓ **Reutilización:** reduce la necesidad de los usuarios de recordar.

El sistema posee una barra de menú con las opciones básicas del sistema, una barra lateral izquierda con las opciones que dispone el administrador, un área donde el usuario visualiza el contenido de lo seleccionado en la barra lateral izquierda (Figura 34) y una barra lateral donde se indica la última fecha de ingreso al sistema. Las notificaciones en caso de alertas se las visualiza en un ícono en la barra de notificaciones.

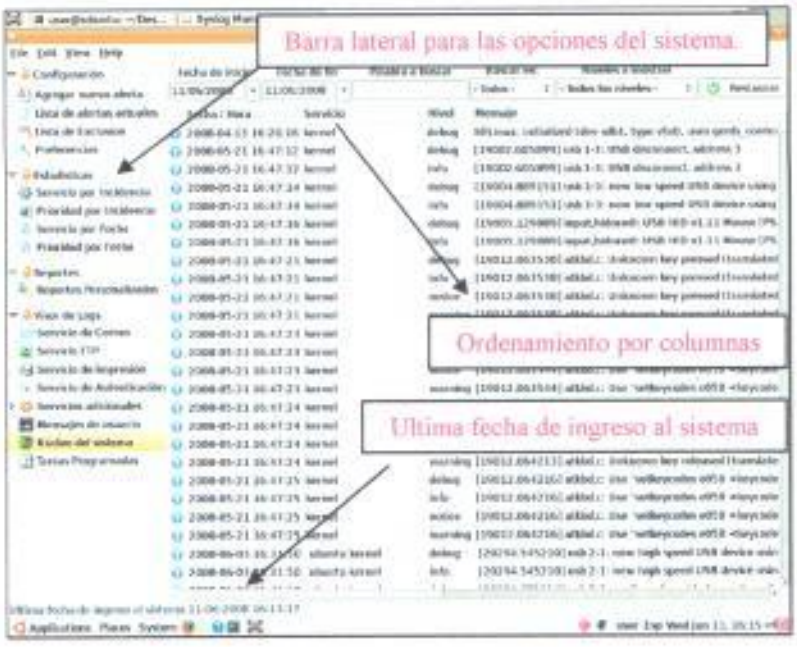

Figura 40. Vista de logs del SyslogManager.

La vista de logs posee una barra de búsqueda, con lo que el usuario podrá filtrar los mensajes de logs al criterio del usuario (por fecha, servicio, nivel o frase dentro del mensaje), disponiendo además de ordenamiento por cada columna de la vista de logs.

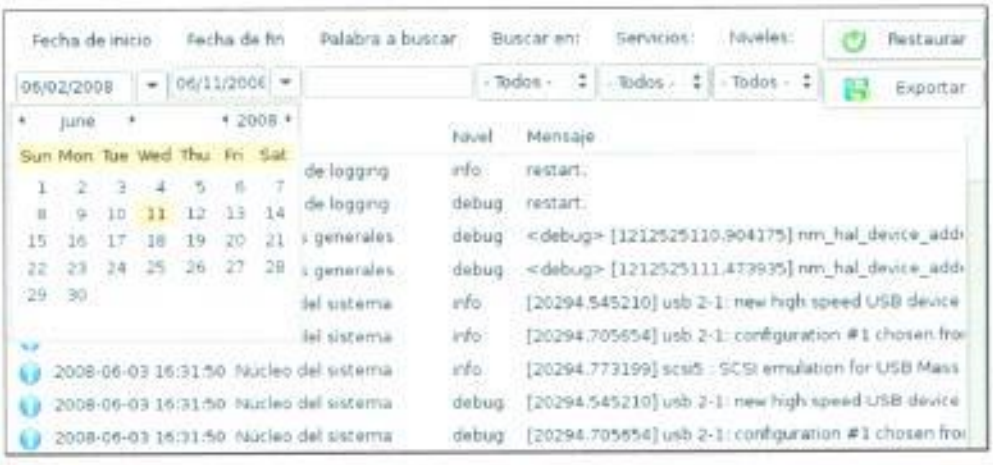

Figura 41. Barra de búsquedas de los mensajes de logs

Para resultados de ingresos de datos o de problemas ocurridos se muestran mensajes indicando el problema o el evento.

Figura 42. Mensaje que aparece al guardar una alerta

Las estadísticas poseen un historial de los datos que retornan, ubicados a la izquierda del gráfico, además de contar con el clic derecho para tener la vista por histogramas o gráfico circular y para guardar la imagen seleccionada.

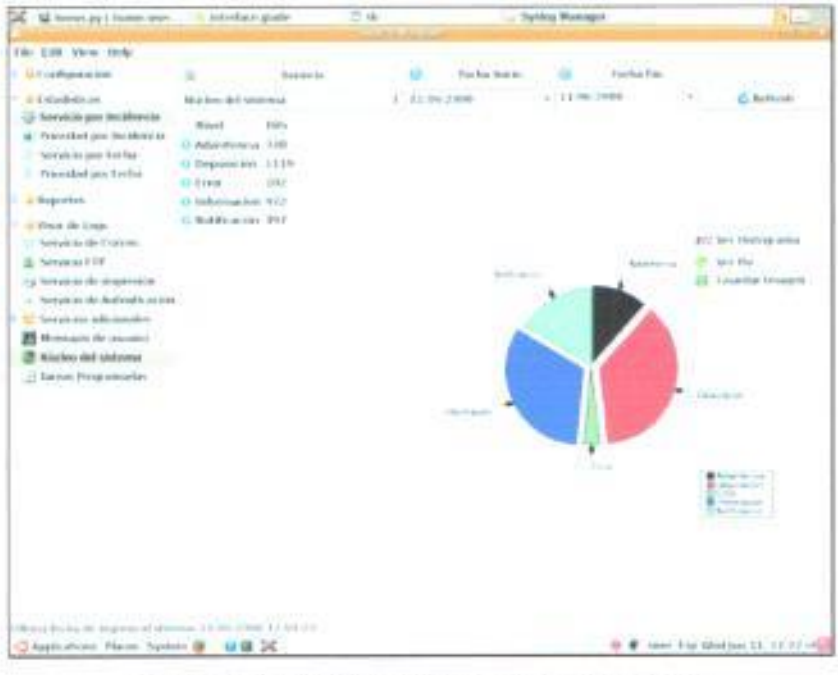

Figura 43. Estadísticas del sistema figura 1.

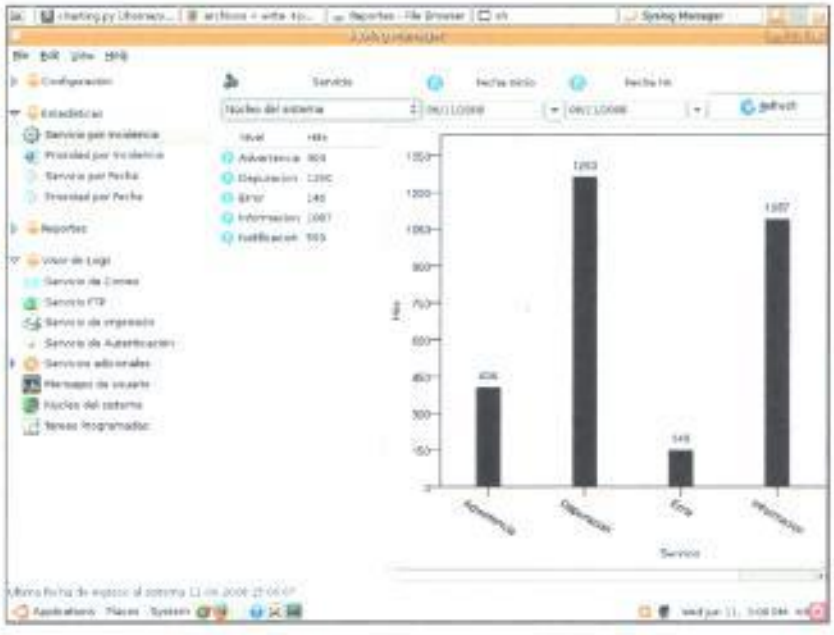

Figura 44. Estadísticas del sistema figura 2.

CAPÍTULO 5

IMPLEMENTACIÓN

5.1. Proceso de implementación

Esta sección incluye una descripción del proceso de implementación de la aplicación demostrativa de este proyecto de tesis, una vez realizadas las etapas de análisis y diseño del sistema.

Se han definido reglas de programación, descritas a continuación:

- Los nombres de las clases y métodos deben ser significativos.
- Si el nombre de una clase involucra una segunda palabra, por convención ésta iniciará con mayúscula. Por ejemplo: wndMain()
- Los nombres de las variables son con minúscula, tanto globales como locales.
- El código debe estar debidamente documentado, para cada clase y método.

Como se indicó en el análisis de tecnologías, el lenguaje de programación utilizado fue Python, que es un lenguaje interpretado y multiplataforma ideal para aplicaciones Linux.

Para el diseño de la interfaz se empleó la herramienta GLADE, que permite crear interfaces GTK y GNOME compatibles con cualquier distribución de Unix.

Para la extracción de datos de los mensajes generados por los logs, se creó un servicio que arranca automáticamente al iniciarse el sistema operativo. Este servicio se encarga de monitorear periódicamente si un evento ha ocurrido, clasificándolo por su prioridad o servicio para su posterior almacenamiento en la base de datos.

Adicionalmente se instaló el IDE KOMODO que posee un ambiente de programación amigable ajustable a la interfaz gráfica diseñada en GLADE. La base de datos utilizada para esta tesis es SQLITE.

Luego de configurar tanto el IDE, la base de datos y el servicio que genera los mensajes de logs, se procedió a la codificación de un módulo de conexión a la base de datos desde una aplicación Python.

Las pruebas de conexión se realizaron usando el IDE KOMODO y mediante el módulo de conexión se verificó el ingreso y la extracción correcta de datos en Python.

Se crearon también módulos para la vista de logs como para la generación de reportes, estadísticas y alarmas, todos desarrollados en Python.

Una vez terminada la lógica de programación, se realizaron pruebas para verificar el correcto funcionamiento del sistema y corregir los errores no detectados.

5.2 Plan de pruebas

Las pruebas son un proceso que permite determinar el correcto funcionamiento de los componentes del sistema, enfocándose en la lógica interna y las funciones externas del sistema con el objeto de descubrir errores.

Las pruebas consistían en instalar el sistema en diversos ambientes Unix, se probó con usuarios de Ubuntu, Fedora y FreeBSD. El objetivo era probar que la recepción de los mensajes a la base de datos sea en tiempos prudentes, además de probar las funcionalidades y emitir comentarios y sugerencias tanto en la funcionalidad como en la apariencia del sistema.

Existen algunos tipos de pruebas importantes dentro del diseño de un software entre ellas se encuentran las pruebas unitarias, las pruebas de integración, las pruebas de aceptación y las pruebas aleatorias [27].

Pruebas Unitarias El objetivo de las pruebas unitarias es probar un módulo de código y tratar de obtener un funcionamiento aceptable de éste. Así, cada una de las clases, procedimientos y funciones se aislaron individualmente para demostrar su correcta aplicación conforme se desarrollaba la aplicación.

Pruebas de Integración: Las pruebas de integración se realizaron una vez aprobadas las pruebas unitarias. Su objetivo fue comprobar el correcto funcionamiento conjunto de las pruebas unitarias, es decir se verificó que las clases y procedimientos realizaban las llamadas correctas una a la otra.

Para realizar estas pruebas, existen dos métodos: estructurales o de caja blanca, y funcionales o de caja negra.

- Las pruebas de caja blanca consisten en probar exhaustivamente la estructura del código fuente, lo cual involucra cobertura de sentencias, ramas, condiciones y bucles.

- Las pruebas de caja negra, en cambio permiten detectar el funcionamiento incorrecto o incompleto, errores de interfaz, errores en accesos a estructuras de datos externas, problemas de rendimiento, errores de inicio y terminación. Su criterio se basa en las interfaces y las especificaciones de los módulos. Es recomendable realizarla con los usuarios.

Pruebas de integridad de los datos: Estas pruebas verifican que los procedimientos y métodos de acceso a la base de datos funcionen correctamente, garantizando la recuperación exacta de los cambios hechos en la base.

Las técnicas empleadas consistieron en llamar a cada procedimiento con datos válidos e inválidos, inspeccionar la base de datos para asegurar que los datos son los previstos y revisar los valores devueltos para asegurar que la recuperación de datos es correcta.

Al realizar estas pruebas se consideró que los procesos deberían invocarse manualmente, y se empleó una base de datos de tamaño pequeño para incrementar la visibilidad de cualquier evento no aceptable. Se empleó la herramienta SQLITEBROWSER para la visualización de los datos.

Pruebas de funcionalidad: Estas pruebas requirieron la ejecución de la secuencia de pasos indicadas en los casos de uso y escenarios descritos en la sección 3.3. Las pruebas fueron realizadas con datos correctos e incorrectos, a fin de comprobar que el flujo resultante era el señalado. Se pudo verificar:

- Obtención de resultados esperados al emplear datos válidos.
- Obtención de mensajes de error o advertencias al emplear datos inválidos.
- Aplicación correcta de la lógica de programación.

Pruebas de interfaz de usuario: Estas pruebas verifican la interacción del usuario con el sistema. Su objetivo consiste en asegurar que la interfaz y flujo de pantallas permitan al usuario acceder y navegar a través de toda la funcionalidad de la aplicación. Estas pruebas verifican:

- Sensación del usuario frente a la aplicación y su apariencia.
- Orientación intuitiva en la aplicación, sin necesidad de acudir a un manual.

- Claridad en nombres de los campos, mensajes de error, estado del sistema.
- Las ventanas y sus características, como menús, tamaño, posición y estado cumplen los estándares.

Pruebas de aceptación: Las pruebas de aceptación se realizan por un grupo de usuarios finales o los clientes del sistema, quienes corroboran que el sistema desarrollado cumple los requisitos planteados [28].

Pruebas Aleatorias: Las pruebas aleatorias consisten en generar modelos, generalmente estadísticos, que representen las posibles entradas al programa para crear a partir de ellos los casos de prueba. Para tal fin se introdujeron grupos de datos de entrada que especificaban servicios o niveles a administrar del Syslog.

Las pruebas iniciales del funcionamiento del sistema arrancaron con la instalación de la correspondiente aplicación para cada una de las distribuciones indicadas. Para ello se siguieron los pasos indicados en los requerimientos en el Anexo C correspondientes al Manual de Usuario.

Las pruebas fueron realizadas por administradores de Unix en las distribuciones Fedora y Ubuntu, se tomo un caso de prueba que consistía en obtener un reporte de todos los mensajes "Núcleo del sistema" con las fechas desde el 4 de mayo del 2008 al 10 de junio del 2008, para ello se hicieron los siguientes pasos descritos en el caso de prueba 0:

CASO DE PRUEBA 0: Usuario genera reportes filtrando por servicio "Núcleo del sistema" y las fechas entre el 4 de mayo del 2008 y 10 de junio del 2008.

Precondiciones: Deben existir registros en la base de datos para que el reporte contenga información.

Descripción: Se debe generar un reporte en XML filtrando por servicio y por fecha, la prueba se la hizo en 2 distribuciones diferentes.

Pasos a Seguir:

1. El administrador ingresó al sistema administrador de logs.
2. Ingresa a Reportes -> Reportes por filtros.
3. Se muestra una barra de filtros por servicio, nivel, fecha y búsquedas por palabras dentro del mensaje de log y una lista con todos los mensajes de logs.
4. Usuario selecciona en la fecha de inicio 4 de mayo del 2008, fecha fin 10 de junio del 2008 y en servicio "Núcleo del sistema".

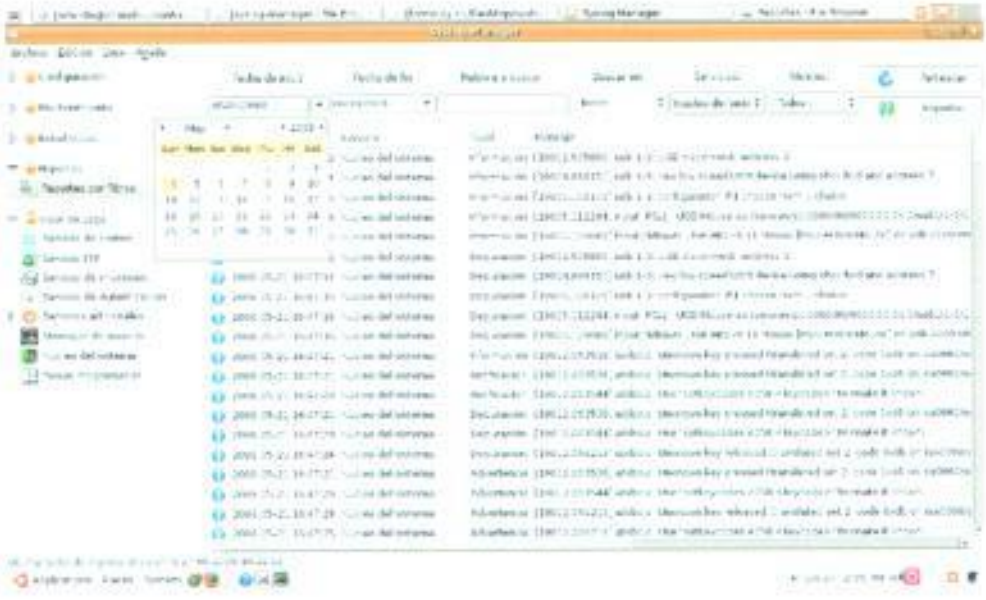

Figura 45. Filtros de la sección reportes.

5. Una lista con las coincidencias es generada.
6. El usuario selecciona Exportar, ingresa la ruta de destino y da clic en "*Exportar*".

Resultados Obtenidos: Dependiendo de la información ingresada en los filtros, se listarán los mensajes que cumplan el criterio ingresado.

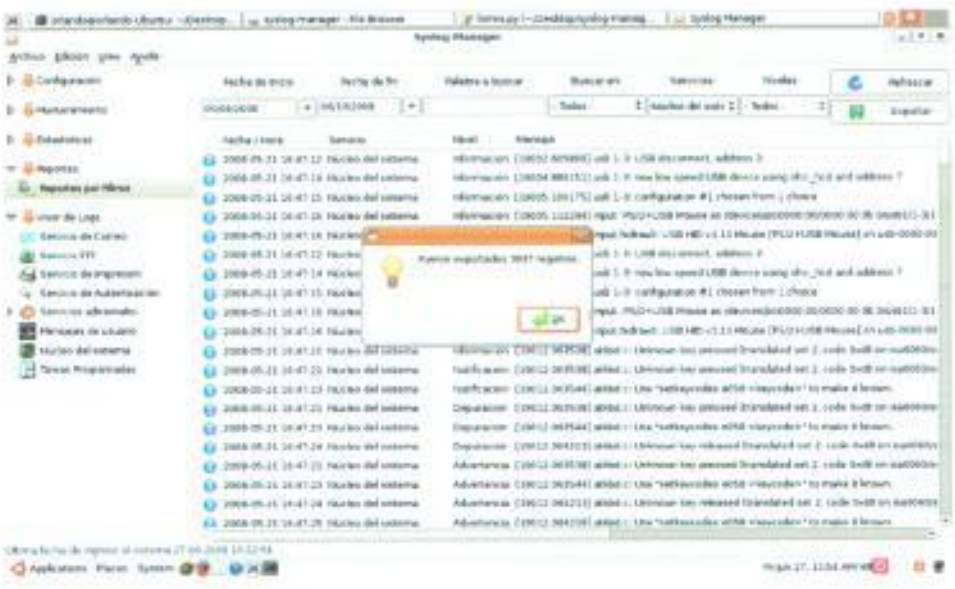

Figura 46. Reporte generado en Ubuntu

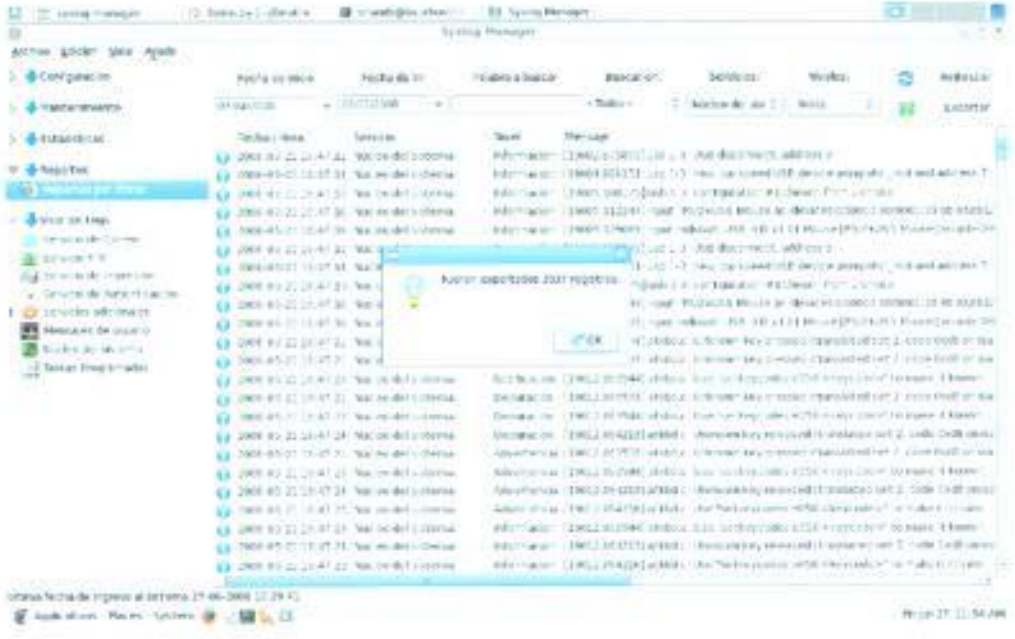

Figura 47. Reporte generado en Fedora 9.

Poscondiciones: Se generará un archivo XML en la ruta ingresada por el usuario, además de un archivo XSLT para poder ser visualizado en un navegador Web.

Figura 48. Reporte generado en XML.

Como el caso descrito, los usuarios de prueba se encargaron de testear la funcionalidad del sistema en los siguientes casos descritos a continuación.

CASO DE PRUEBA 1: Usuario genera alertas

Precondiciones: La alerta no debe haber sido creada previamente por el administrador.

Descripción: El administrador desea programar mediante el sistema *SyslogManager* el envío de una alerta hacia su correo electrónico para un servicio y nivel específico.

Pasos a Seguir:

1. El administrador ingresó al sistema administrador de logs.
2. Ingresa a Configuración -> Agregar nueva alerta
3. Se muestra un formulario al administrador donde debe llenar los datos necesarios para generar la alarma. Éstos son:

- **Servicio:** Servicio de Correo
- **Nivel:** Alerta
- **Destino:** Dirección de Correo
- **Detalles del destino:** ocrespo@vgl.cl
- **Observaciones:** Ninguna

4. Da clic en el botón *"Aceptar"*.

Resultados Obtenidos: Aparece un mensaje indicando "Alerta ingresada exitosamente". Los datos se almacenan en la base de datos para su uso posterior.

Poscondiciones: Esta alarma se disparará cuando un mensaje en el log con las características señaladas se presente. Además se agregará a la lista de alertas en la opción "Configuración".

CASO DE PRUEBA 2: Usuario genera estadísticas

Precondiciones: Deben existir logs creados previamente.

Descripción: El Administrador desea observar las estadísticas por servicio.

Pasos a Seguir:

1. El administrador ingresó al sistema administrador de logs.
2. Ingresa a Estadísticas -> Servicio por incidencia.
3. Se muestra una barra de filtros, tanto por servicio como por fecha.
4. Usuario selecciona el filtro deseado y automáticamente se genera la estadística.

Resultados Obtenidos: Dependiendo de la información ingresada en los filtros, se generará el gráfico estadístico al usuario, en caso de no poseer información se presentará un mensaje indicando "gráfico no disponible".

Poscondiciones: Se genera gráfico estadístico al usuario.

CASO DE PRUEBA 3: Usuario genera reportes por filtros.

Precondiciones: Deben existir logs creados previamente.

Descripción: El administrador desea generar reportes y exportarlos como XML.

Pasos a Seguir:

7. El administrador ingresó al sistema administrador de logs.
8. Ingresa a Reportes -> Reportes por filtros.
9. Se muestra una barra de filtros por servicio, nivel, fecha y búsquedas por palabras dentro del mensaje de log y una lista con todos los mensajes de logs.
10. Usuario selecciona el filtro deseado y la lista de mensajes se actualiza dependiendo del criterio ingresado.
11. El usuario selecciona Exportar, ingresa la ruta de destino y da clic en "*Exportar*".

Resultados Obtenidos: Dependiendo de la información ingresada en los filtros, se listarán los mensajes que cumplan el criterio ingresado.

Poscondiciones: Se generará un archivo XML en la ruta ingresada por el usuario, además de un archivo XSLT para poder ser visualizado en un navegador Web.

CASO DE PRUEBA 4: Usuario crea una configuración personalizada al *syslog.conf*.

Precondiciones: Deben existir logs creados previamente.

Descripción: Seleccionando los servicios y los niveles, el usuario configura el *syslog.conf*.

Pasos a Seguir:

1. El administrador ingresó al sistema administrador de logs.
2. Ingresa a Configuración -> Configurar Syslog.
3. Se muestran las opciones por configuración original o personalizada, el usuario escoge personalizada.
4. Usuario selecciona servicio y nivel y da clic en "*Agregar*", una lista con las opciones escogidas se va llenando.
5. Cuando todas las selecciones han sido escogidas, da clic en "*Finalizar*".

Resultados Obtenidos: La configuración del *syslog.conf* será reemplazada por la información seleccionada por el usuario.

Poscondiciones: El servicio que intercepta los mensajes de logs, solamente mostrará al usuario las combinaciones de servicio/nivel seleccionadas.

CASO DE PRUEBA 5: Usuario modifica la configuración personalizada del *syslog.conf*.

Precondiciones: Deben existir logs creados previamente y una configuración personalizada creada.

Descripción: Usuario modifica la configuración personalizada creada en una instancia anterior.

Pasos a Seguir:

1. El administrador ingresó al sistema administrador de logs.

2. Ingresa a Configuración -> Configurar Syslog.
3. Se muestran las opciones por configuración original o personalizada. El usuario escoge personalizada.
4. Usuario visualiza la configuración anterior en la lista de configuraciones seleccionadas por el usuario.
5. El usuario tiene la opción de agregar o quitar configuraciones mediante los botones diseñados para tal fin. Así podrá seleccionar el servicio y nivel y dar clic en "*Agregar*", para remover una configuración previa, debe seleccionarla de la lista inferior y dar clic en "*Quitar*".
6. Cuando todas las selecciones han sido escogidas, dar clic en "*Finalizar*".

Resultados Obtenidos: La configuración del *syslog.conf* será reemplazada por la información seleccionada por el usuario.

Poscondiciones: El servicio que intercepta los mensajes de logs, solamente mostrará al usuario las combinaciones de servicio/nivel seleccionadas.

CASO DE PRUEBA 6: Usuario envía mensaje de log a la lista de exclusión

Precondiciones: Deben existir logs creados previamente.

Descripción: Usuario selecciona un mensaje de log y lo envía a la lista de exclusión.

Pasos a Seguir:

1. El administrador ingresó al sistema administrador de logs.
2. Ingresa a "*Visor de Logs*" y selecciona algún servicio de log.
3. Se listan todos los mensajes de logs del servicio seleccionado.

4. Usuario da clic derecho sobre uno de los mensajes de logs y selecciona "*Enviar a lista de exclusión*".

Resultados Obtenidos: El mensaje de log seleccionado (servicio, mensaje) pasa a la lista de mensajes excluidos de la visualización de logs.

Poscondiciones: Los mensajes excluidos no serán visualizados.

CASO DE PRUEBA 7: Usuario remueve un mensaje de la lista de exclusión.

Precondiciones: Deben existir logs creados previamente y mensajes en la lista de exclusión.

Descripción: Usuario quita un mensaje de la lista de exclusión.

Pasos a Seguir:

1. El administrador ingresó al sistema administrador de logs.
2. Ingresa a Configuración -> Lista de Exclusión.
3. Se listan todos los mensajes existentes en la lista de exclusión.
4. Usuario da clic derecho sobre uno de los mensajes de logs y selecciona "*Quitar*".

Resultados Obtenidos: El mensaje de log seleccionado (servicio, mensaje) es quitado de la lista de exclusión.

Poscondiciones: Los mensajes removidos de la lista de exclusión volverán a visualizarse entre los mensajes de logs.

CASO DE PRUEBA 8: Usuario respalda mensajes de logs.

Precondiciones: Deben existir logs creados previamente.

Descripción: Usuario respalda mensajes de logs basándose en un criterio (fecha, servicio o nivel).

Pasos a Seguir:

1. El administrador ingresó al sistema administrador de logs.
2. Ingresa a Mantenimiento -> Respaldar logs.
3. Usuario indica el nombre y el destino del archivo de respaldo.
4. Usuario escoge filtros para el respaldo, sean estos por fecha inicio-fecha fin, servicio o nivel.
5. Usuario da clic en "*Respaldar*".

Resultados Obtenidos: El archivo de respaldo es creado a partir de los filtros indicados en el destino indicado por el usuario. El sistema muestra el mensaje "Fueron respaldados # registros"

Poscondiciones: El usuario tendrá un respaldo de los mensajes indicados para un análisis particular de ellos.

CASO DE PRUEBA 9: Usuario restaura mensajes de logs.

Precondiciones: Debe existir un archivo de respaldo válido.

Descripción: Usuario restaura mensajes de logs que fueron previamente respaldados.

Pasos a Seguir:

1. El administrador ingresó al sistema administrador de logs.
2. Ingresa a Mantenimiento -> Restaurar logs.
3. El usuario debe escoger la ruta donde se encuentra el archivo con el respaldo.

4. Usuario da clic en "*Restaurar*".
5. Un mensaje de archivo restaurado es mostrado al usuario.

Resultados Obtenidos: Los mensajes son agregados a la base de datos.

Poscondiciones: Los mensajes contenidos dentro del respaldo se visualizarán en el visor de logs.

CASO DE PRUEBA 10: Usuario elimina mensajes de logs.

Precondiciones: Deben existir logs creados previamente.

Descripción: Usuario elimina mensajes de logs.

Pasos a Seguir:

1. El administrador ingresó al sistema administrador de logs.
2. Ingresa a Mantenimiento -> Eliminar logs.
3. Una pantalla es mostrada al usuario, donde contiene una serie de filtros necesarios para eliminar los logs.
4. Usuario da clic en "*Eliminar*".

Resultados Obtenidos: Los mensajes de logs son eliminados de la base de datos y un mensaje indicando la eliminación exitosa de éstos es presentado al usuario.

Poscondiciones: Aquellos mensajes que cumplían con el criterio de eliminación no mostrarán más en las vistas de logs, debido a que fueron eliminados de la base de datos.

5.3 Resultados de las pruebas

Como resultado de los casos de prueba vistos en la sección 5.2, se efectuaron las siguientes modificaciones al sistema:

- Debido a los problemas relacionados con dependencias, se optó por trabajar con librerías gráficas que fueran compatibles con todas las distribuciones empleadas en esta tesis y cuya configuración fuera lo más sencilla posible.

- Se crearon validaciones para alarmas repetidas, gráficos estadísticos donde no se encontró coincidencias en los filtros, etc.

- Los filtros de mensajes de logs se los hizo sobre la misma ventana que visualiza el mensaje, debido a que antes se llamaba a una forma para que hiciera los filtros.
- Se crearon nombres estándar para las vistas de logs, evitando nombres que el usuario no pueda interpretar. De igual forma se procedió para nombrar los servicios y niveles.

- Se dió seguimiento a las transacciones del sistema, a fin de comprobar la veracidad e integridad de los datos almacenados.

- Se incorporó la opción de guardar los gráficos estadísticos en formato PNG, para el gráfico circular (pie chart), histogramas y líneas de tiempo.

- Con el propósito de añadir seguridad al sistema, se incluyó la "Última Fecha de ingreso al Sistema" como parte de la barra de estado. Este mensaje actúa como un recordatorio para el administrador y mediante el cual puede determinar si alguien más está usurpando su identidad.

CONCLUSIONES Y RECOMENDACIONES

En esta sección se presentan las conclusiones obtenidas en el desarrollo e implementación del proyecto. Además, recomendaciones para el uso de código libre y su uso en diversos ambientes Unix.

- El uso de esta herramienta ayudará a los administradores de sistemas Unix a llevar un mejor control gracias a los reportes y estadísticas que este brinda, además de las alarmas que son necesarias para advertir en caso de algún inconveniente en el sistema.

- Existen muchos lenguajes de código libre en el ambiente Unix tales como C, QT, Python que cada vez toman más renombre debido a su facilidad de implementación, gran cantidad de foros de ayuda y documentación a través de la Web, muchos de estos lenguajes están siendo utilizados en ambientes Windows e incluso ambientes Web.

- Debido a que el sistema está implementado sobre diversos ambientes Unix, el uso de máquinas virtuales facilitó las pruebas sobre cada uno de estos ambientes. La disponibilidad de estos sistemas operativos fue fácilmente descargable desde el sitio web www.vmware.com, donde existe una sección para descargar las últimas actualizaciones de cada distribución.

- El uso de librerías graficas para las estadísticas resultó un proceso de análisis, debido a que las librerías utilizadas debían ser fácilmente instalables para cualquiera de las distribuciones de prueba de esta tesis y sobre todo libre de dependencias, lo que hace al software más portable.

- Se pasó por muchas pruebas para seleccionar el lenguaje de programación a utilizar en este proyecto de tesis, debido a que debía ser portable, existente en todas las distribuciones y sobre todo ajustable a librerías externas como es el caso de las librerías gráficas.

- Para nosotros como desarrolladores, fue nuestra primera experiencia con código libre y se llega a la conclusión de que no hay mucho que envidiar con el software comercial. Además posee el soporte de miles de personas que día a día crean nuevos códigos, nuevas mejoras e incluso modificaciones a software existentes.

- Para algunas distribuciones de Unix, su instalación fue más complicada que para aquellas con más renombre como Fedora o Ubuntu, tal es el caso de FreeBSD que al principio resultó muy complicada su instalación; pero gracias a la gran cantidad de documentación existente se pudo instalar una distribución funcional y operativa.

- Al hacer las comparaciones con software existente para el análisis y monitoreo de logs en Unix, se encontró que muchos de ellos disponían de funciones básicas e incluso utilizaban la consola como parte de su interfaz. Otras, cuya interfaz era más amigable estaban disponible en la red por un precio por su desarrollo, se espera que esta herramienta ayude a los administradores y facilite el monitoreo de los logs.

- Se debe promover la enseñanza de código abierto en las instituciones educativas de nivel superior con el fin de poder desarrollar software que pueda hacer competencia a software comercial. Además, se podrían crear comunidades internas que permitan compartir ideas y mejoras para códigos existentes.

BIBLIOGRAFÍA

1. Julio Pérez - Instituto de Ingeniería Eléctrica. *Syslog y Archivos de registro de eventos*. Disponible en web: <http://iie.fing.edu.uy/ense/asign/admunix/logs.htm>.

2. Revista PC paso a paso edición 29, escrito por Iván Alcaraz. *Control de logs en GNU/Linux*, actualizado junio 2005. Disponible en formato PDF en: <akira.azul.googlepages.com/logs.pdf>.

3. Network Administrators Survival Guide - Chapter 4. *Using syslog*, actualizado septiembre 2005. Disponible en formato PDF en: <http://www.informit.com/content/images/1587052113/samplechapter/1587052113content.pdf>.

4. LOGSURFER HOMEPAGE. *Sitio oficial logsurfer* [en línea]. Disponible en Web:<http://www.dfn-cert.de/eng/logsurf/>.

5. LOGWATCH HOMEPAGE. *Sitio oficial logwatch* [en línea]. Disponible en Web: <http://www2.logwatch.org:81/>.

6. Debian - Logwatch. *Analizador de registros de salida agradable escrito en Perl*, actualizado periódicamente. Disponible en Web: <http://packages.debian.org/sid/logwatch?lang=es>.

7. LOGCHECK HOMEPAGE. *Sitio oficial logcheck* [en línea]. Disponible en Web: <http://logcheck.org/>.

8. Gerardo Fernández Navarrete, *Detección de Ataques*, actualizado 2002, Disponible en Web: <http://gerardo.info/papers/deteccion.php>.

9. SWATCH HOMEPAGE. *Sitio oficial swatch* [en línea]. Disponible en Web: <http://linux.maruhn.com/sec/swatch.html>.

10. PSIONIC LOGCHECK HOMEPAGE. *Sitio oficial* Psionic Logcheck [en línea]. Disponible en Web: <http://www.psionic.com/abacus/logcheck/>.

11. Kurt Seifried. *Guía de Seguridad del Administrador de Linux* [en línea], actualizado Febrero 2006. Disponible en Web: <http://www.wikilearning.com/tutorial/guia_de_seguridad_del_administrador_de_linux-ficheros_de_log_y_otros_metodos_de_monitorizacion/9634-50>

12. COLORLOGS HOMEPAGE. *Sitio oficial* colorlogs [en línea]. Disponible en Web: <http://www.resentment.org/projects/colorlogs/>.

13. Linux Server Security (2da. Edición), Editorial O'Reilly. *System Log Management and Monitoring*, actualizado 2005. Disponible en formato PDF en: <http://www.delmar.edu/Courses/ITSY2401/eBooks/Log_monitoring_and_management.pdf>.

14. Wikipedia: *Syslog*, actualizado 30 de Enero del 2008. Disponible en Web: <http://es.wikipedia.org/wiki/Syslog>.

15. Celso González. *Metalog, la herramienta para nuestros logs*, actualizado 20 de Enero del 2005. Disponible en Web: <http://www.emagister.com/metalog-herramienta-para-nuestros-logs-cursos-1103132.htm#programa>.

16. Linux Magazine. *Registro de Sistema de Próxima Generación: Syslog-NG*, actualizado 22 de Noviembre del 2004. Disponible en formato PDF en: <http://www.linux-magazine.es/Issue/01/Syslog.pdf>.

17. Martin Méndez. *El sistema Unix*, actualizado 23 de Febrero del 2007. Disponible en Web: <http://www.infranetworking.net/foros/lofiversion/index.php/t453.html>.

18. Comunidad Ubuntu. *Ubuntu: GNU/Linux para seres humanos*, actualizado 11 de Octubre del 2007. Disponible en Web: <http://www.ubuntu-es.org/ubuntu/introduccion>.

19. FEDORA PROJECT. Sitio oficial Fedora [en línea]. Disponible en Web: <http://fedoraproject.org/wiki/Overview>.

20. FREEBSD HOMEPAGE. Sitio oficial Freebsd [en línea]. Disponible en Web: <http://www.freebsd.org/es/>.

21. Wikipedia. *MAC OS X 10.4* Disponible en Web: <http://es.wikipedia.org/wiki/Mac_OS_X_v10.4>.

22. Wikipedia. *Lenguaje de Programación C, actualizado* 15 de Febrero del 2008. Disponible en Web: <http://es.wikipedia.org/wiki/ANSI_C>.

23. Desarrolloweb. *Que es Perl*, *Disponible en Web*: <http://www.desarrolloweb.com/articulos/541.php>.

24. Wikipedia. *Lenguaje AWK*. Disponible en Web: <http://es.wikipedia.org/wiki/AWK>.

25. Desarrolloweb. *Que es Python*. Disponible en Web: <http://www.desarrolloweb.com/articulos/1325.php>.

26. Alzado.org. Usabilidad sin usuarios: Heurística. Disponible en web: <http://www.alzado.org/articulo.php?id_art=221>

27. Ingeniería del software. Pruebas del Software. Disponible en formato PDF en: <http://ji.ehu.es/mikelv/index_archivos/ApuntesIS/8-PruebasdelSoftware.pdf>

28. Guía de Técnicas. Pruebas. Disponible en Web: http://www.lpsi.eui.upm.es/MDes/TfcMetrica/GTPrueb.htm

29. RFCS 3164: The BSD Syslog Protocol. Disponible en Web: <http://www.faqs.org/rfcs/rfc3164.html>.

ANEXOS

ANEXO A

GLOSARIO

Administrador del Sistema: Es aquella persona que se dedica a mantener y operar un sistema de cómputo ó una red.

Autenticaciones: Procedimiento de comprobación de la identidad de un usuario. Mediante el mismo se garantiza que el usuario que accede a un sistema de ordenador es quién dice ser. Por lo general, los sistemas de autenticación están basados en el cifrado mediante una clave o contraseña privada y secreta que sólo conoce el auténtico emisor.

Base de Datos: Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.

BCPL: Lenguaje de Programación Básico Combinado creado en 1966.

Comandos: Es una instrucción o mandato que el usuario proporciona a un sistema informático, desde la Consola de comandos o desde una llamada de programación.

Consola de Comandos: Es un programa informático que actúa como Interfaz de usuario para comunicar al usuario con el sistema operativo mediante una ventana que espera ordenes escritas por el usuario en el teclado.

Cron: Es un administrador regular de procesos en segundo plano (demonio) que ejecuta programas a intervalos regulares, es un programador de tareas para Unix.

Crontab: Contiene información sobre los eventos que se realizaran en el sistema cada cierto tiempo (hora, día, semana, mes).A diferencia del directorio */var/spool/cron* donde cada usuario tiene su archivo y especifica sus horarios, este archivo mantiene un crontab que ejecuta los archivos que se encuentren en los directorios *cron.hourly, cron.daily, cron.weekly, cron.monthly*.

Demonio (Servicio): Es un tipo especial de proceso informático que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo).

Destino: Es la secuencia de un directorio o archivo (default: */var/log*) o */dev/console* la consola.

Encriptamiento: Es usado para asegurar la seguridad en los sistemas distribuidos. El esquema más apropiado es que cuando 2 entidades se quieren comunicar establecen una clave de comunicación para ayudar a una autentificación del servidor.

Ficheros binarios: Un fichero binario, contrariamente a un fichero ASCII, contiene más que simplemente texto. Puede contener fotos, sonido, hojas de cálculo, o documentos concebidos para el procesamiento de texto. Los ficheros binarios están formados de unos y ceros.

Fichero FIFO (Tuberías con nombre): Las tuberías o "pipes" simplemente conectan la salida estándar de un proceso con la entrada estándar de otro.

Ficheros texto plano: Son aquellos que están compuestos únicamente por texto sin formato, sólo caracteres.

FTP: Es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor.

IMAP: Es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

Intérprete: Un intérprete es un programa capaz de analizar y ejecutar otros programas.

Kernel: Es la parte fundamental de un sistema operativo. Es el software responsable de facilitar a los distintos programas acceso seguro al hardware.

Klogd: Es un demonio del sistema que permite interceptar los mensajes del Kernel.

Lenguaje ABC: es un lenguaje de programación de alto nivel, creado originalmente como sustituto del BASIC.

Lenguaje de Programación: Es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Lenguaje interpretado: Es un lenguaje de programación que fue diseñado para ser ejecutado por medio de un intérprete.

Linux: Es un sistema operativo que fue creado al fusionar las utilidades y librerías del proyecto GNU con el Kernel de Linux.

Logs o Registro de Eventos: Un log es un registro de actividad de un sistema, que generalmente se guarda en un fichero de texto, al que se le van añadiendo líneas a medida que se realizan acciones sobre el sistema.

Prioridad (Priority): Donde puede ser *debug, info, notice, warning, err, crit, alert, emerg, none.*

ROOT: Es el nombre convencional de la cuenta de usuario que posee todos los derechos en todos los modos (mono o multiusuario). ROOT es también llamado superusuario.

Selector: Se lo conoce como al par facility (servicio) – priority (prioridad). Cuando un programa envía un mensaje al syslog, lo hace especificando un valor de "facility" y un valor de "level".

Servidor DNS: (Domain Name System) se utiliza para proveer a las computadoras de los usuarios (clientes) un nombre equivalente a las direcciones IP.

Servidor SMTP: Protocolo que permite recibir y enviar correo.

Servicio (Facility): Donde puede ser *auth, authpriv, cron, daemon, kern ,lpr, mail, news, syslog, user, uucp y local0-local7 (Explicados en el capítulo 2)*

Sistemas operativos: Es un programa o conjunto de programas de computadora destinado a permitir una gestión eficaz de sus recursos.

Sniffer: Es un programa de captura de las tramas de red. Generalmente se usa para gestionar la red.

SSH: Sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos.

Syslogd: Es el encargado de guardar informes sobre el funcionamiento de la máquina. Recibe mensajes de las diferentes partes del sistema (núcleo, programas, etc.) y los envía y/o almacena en diferentes localizaciones.

Syslog.conf: Indica donde deben de ser enviados mensajes del sistema, sus líneas son de la forma: servicio – prioridad – destino.

TCP: Es un protocolo de control de transmisión de datos que garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron.

UDP: Es un protocolo del nivel de transporte basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera.

Unix: Es un sistema operativo portable, multitarea y multiusuario; desarrollado, en principio, en 1969 por un grupo de empleados de los laboratorios Bell de AT&T.

XML: es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

ANEXO B

DICCIONARIO DE DATOS

TABLAS DEL SISTEMA

Nombre de la Tabla	Tipo de Tabla	Clave Primaria	#Columnas
FACILITIES	Independiente	Id	3
LEVELS	Independiente	Id	3
MESSAGES	Dependiente	Id	7
ALERTS	Dependiente	Id	7
LISTAEXCLUSION	Dependiente	Id	3
LOGIN	Independiente		1

TABLA FACILITIES

Almacena información sobre los servicios del sistema (demonios).

COLUMNAS:

Clave	Nombre	Tipo de Dato	Nulo	Descripción
PK	Id	Integer	No	Identificador del servicio.
	Name	Varchar(25)	No	Nombre del servicio
	Description	Varchar(100)	No	Información acerca del servicio.
	Title	Varchar(100)	No	Titulo a Mostrar entendible al usuario.

RELACIONES:

Nombre	Tabla Padre	Tabla Hija	Cardinalidad
Posee	MESSAGES	FACILITIES	1:M

Se_subdivide	LISTAEXCLUSION	FACILITIES	1:M
Se_subdivide	ALERTS	FACILITIES	1:M

INDICES:

Nombre	Columnas	Único
Ind_facilityid	Id	SI

TABLA LEVELS

Almacena información acerca de las prioridades de un servicio.

COLUMNAS:

Clave	Nombre	Tipo de Dato	Nulo	Descripción
PK	Id	Integer	No	Identificador de prioridad.
	Name	Varchar(25)	No	Nombre de la prioridad
	Description	Varchar(100)	Si	Información acerca de la prioridad.
	Severity	Integer	No	Identificador para la severidad de la prioridad
	Title	Varchar(100)	No	Titulo entendible presentado al usuario.

RELACIONES:

Nombre	Tabla Padre	Tabla Hija	Cardinalidad
Esta_asociada	MESSAGES	LEVELS	1:M
Se_subdivide	ALERTS	LEVELS	1:M

INDICES:

Nombre	Columnas	Unico
Ind_levelid	Id	SI

TABLA MESSAGES

Almacena información acerca de los mensajes de log.

COLUMNAS:

Clave	Nombre	Tipo de Dato	Nulo	Descripción
PK	Id	Integer	No	Identificador del mensaje.
FK	Facility	Integer	No	Identificador del servicio
FK	Level	Integer	No	Identificador de la prioridad.
	Date	Date	No	Fecha de generación del log
	Daemon	Varchar(50)	No	Demonio que genera el mensaje
	Host	Varchar(50)	No	Maquina que genero el mensaje.
	Message	Varchar(200)	No	Mensaje del log.

RELACIONES:

Nombre	Tabla Padre	Tabla Hija	Cardinalidad
Posee	MESSAGES	FACILITIES	1:N
Esta_asociada	MESSAGES	LEVELS	1:N

INDICES:

Nombre	Columnas	Unico
Ind_messagemessage	Message	No
Ind_messagefacility	Facility	No
Ind_messagelevel	Level	No

TABLA ALERT

Almacena información acerca de las alertas del sistema.

COLUMNAS:

Clave	Nombre	Tipo de Dato	Nulo	Descripción
PK	Id	Integer	No	Identificador de la alerta.
	Username	Varchar2(50)	No	Usuario que creó la alerta.
FK	Service	Integer	No	Identificador del servicio.
FK	Level	Integer	No	Identificador de la prioridad.
	Destination	Varchar(50)	No	Tipo de destino de la alerta.
	Destination_details	Varchar(100)	No	Detalle del destino.
	Comments	Varchar(200)	Yes	Comentarios sobre la alerta creada.

RELACIONES:

Nombre	Tabla Padre	Tabla Hija	Cardinalidad
Se_subdivide	ALERT	FACILITIES	1:N
Se_subdivide	ALERT	LEVELS	1:N

INDICES:

Nombre	Columnas	Unico
Ind_alertservice	Service	No
Ind_alertLevel	Level	No

TABLA LISTAEXCLUSION

Almacena información acerca de los mensajes que el usuario excluye en la visualización de los mensajes de logs.

COLUMNAS:

Clave	Nombre	Tipo de Dato	Nulo	Descripción
PK	Id	Integer	No	Identificador de la alerta.
FK	Service	Integer	No	Identificador del servicio.
	Message	Varchar(200)	No	Mensaje del log.

RELACIONES:

Nombre	Tabla Padre	Tabla Hija	Cardinalidad
Se_subdivide	LISTAEXCLUSION	FACILITIES	1:N

INDICES:

Nombre	Columnas	Unico
Ind_exclusionmessage	Message	No
Ind_exclusionservice	Service	No

ANEXO C

MANUAL DE ADMINISTRADOR

Requerimientos del Sistema

El usuario de la aplicación, requiere ingresar a la Interfaz gráfica de GNU Linux para poder levantar el sistema *syslog-manager*. Esta herramienta puede ser ejecutada tanto en KDE como GNOME indistintamente.

INGRESO AL SISTEMA

Acceso al Sistema

El administrador de Linux puede acceder al sistema a través del terminal ingresando ***./syslogmanager*** o dando clic en el icono ubicado en el escritorio.

Una vez ingresado al sistema se dispone de módulos para la administración de los logs tal como lo indica la figura C.1.

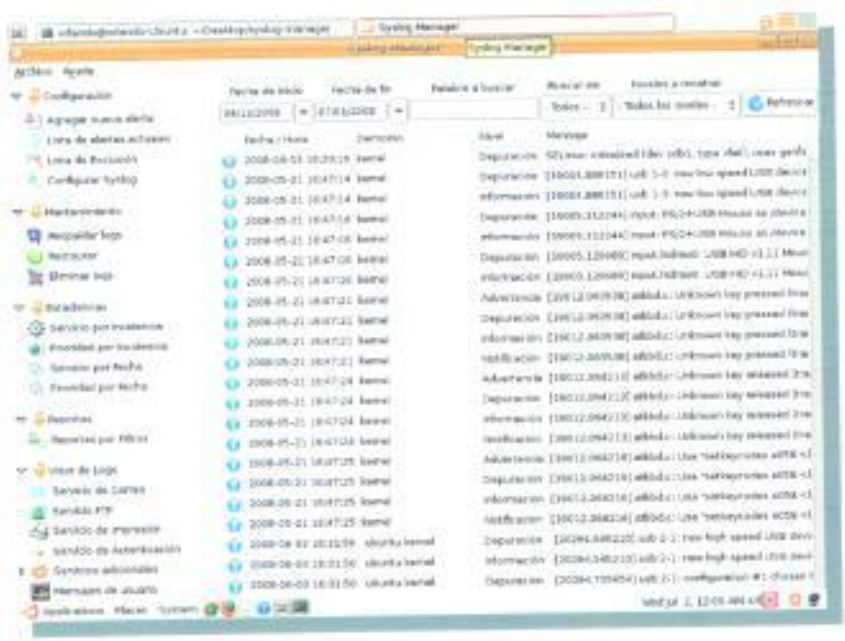

Figura C. 1 Pantalla principal del sistema.

El usuario dispone de módulos los cuales serán explicados paso a paso y con detalle mas adelante, estos módulos se clasifican en:

- ✓ Configuración.
- ✓ Mantenimiento.
- ✓ Estadísticas.
- ✓ Reportes.
- ✓ Visor de logs.

Además se incluye en la barra de estado la fecha de última de ingreso al sistema.

MODULOS DEL SISTEMA

MÓDULO CONFIGURACIÓN

Este módulo permite al administrador configurar el *syslog.conf*, es decir permite configurar que se desea administrar (figura C.2), también permite configurar las alarmas las cuales ayudarán a controlar el funcionamiento del sistema. Entre las opciones de este módulo disponemos de:

- ✓ Agregar nueva alerta.
- ✓ Lista de alertas actuales.
- ✓ Lista de Exclusión.
- ✓ Configurar Syslog.

Figura C. 2. Módulo de configuración.

- **Agregar nueva alerta**

Esta opción permite al administrador del sistema generar alertas cuando ocurra un evento según el tipo de servicio y nivel al cual este pertenezca. También nos permite escoger la forma en la cual deseamos que sea emitida esta alerta; ya sea a un correo, un número celular o como una notificación del sistema. Esta última se muestra como ventana de mensaje en la barra de tareas de Linux en el momento exacto en que ocurra el evento. En la siguiente figura podemos apreciar la creación de una alerta a ser enviada a un número celular (figura C.3).

Figura C. 3. Agregar nueva alerta a enviarse a número celular.

- **Listas de alertas actuales**

En esta sección podemos ver un listado de todas las alertas que el administrador ha creado para el sistema (figura C.4), este listado puede ser filtrado por cualquier de los campos que la conforman.

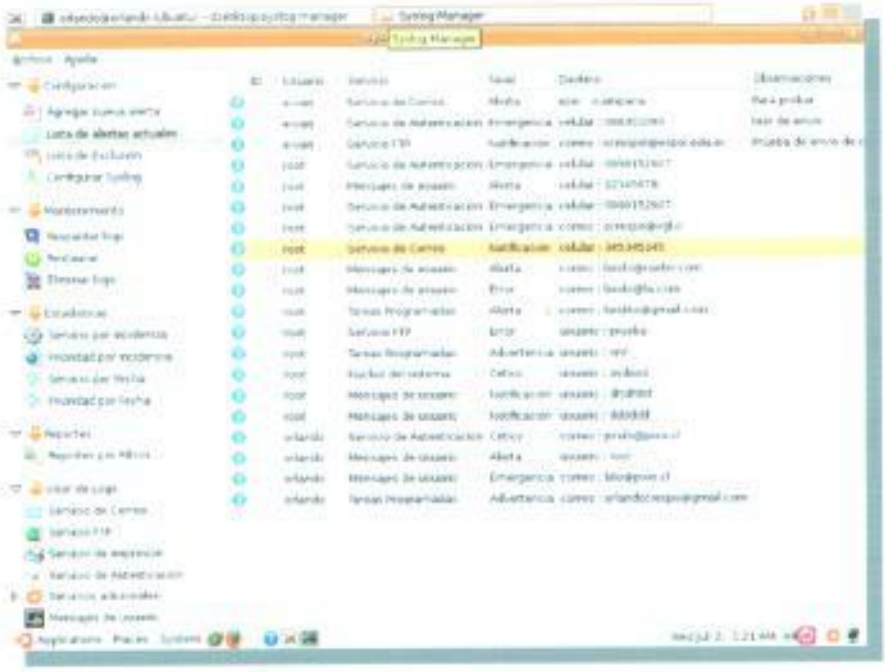

Figura C. 4. Lista de alertas actuales.

Dentro de esta lista de alertas el usuario dispone de dos opciones las cuales se activan al dar clic derecho sobre alguna de las alertas, permitiendo al usuario agregar o eliminar alertas según su criterio (figura C.5).

Figura C. 5. Opciones de la lista de alertas.

Lista de Exclusión

Otro listado al cual tenemos acceso en nuestro sistema es el de los mensajes de logs excluidos (figura C.6); es decir, mensajes que el administrador considere innecesarios de administrar. Estos mensajes de logs no serán mostrados en el visor de log explicado más adelante.

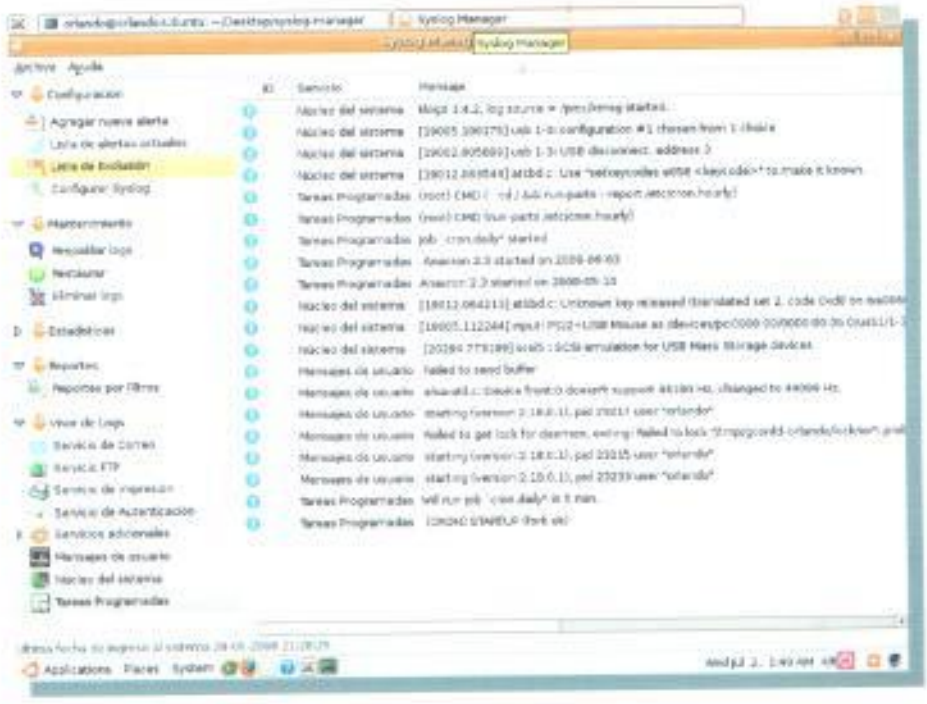

Figura C. 6. Lista de exclusión.

Dando clic derecho sobre uno de los mensajes en la lista de exclusión, se puede eliminar un mensaje que se encuentra excluido. Este mensaje será nuevamente visualizado en el Visor de logs.

Configurar Syslog

Este módulo permite al administrador configurar qué desea administrar, dispone de dos opciones, la configuración original y la personalizada (figura C.7). La configuración original permite administrar el *syslog.conf* como viene predeterminado en el sistema operativo, mientras que la configuración personalizada (figura C.8) permite al administrador seleccionar

la combinación servicio/nivel que desea administrar y la cual será visualizada en el Visor de logs. Esta selección se puede observar en el siguiente gráfico.

Figura C. 7. Configuración del Syslog.

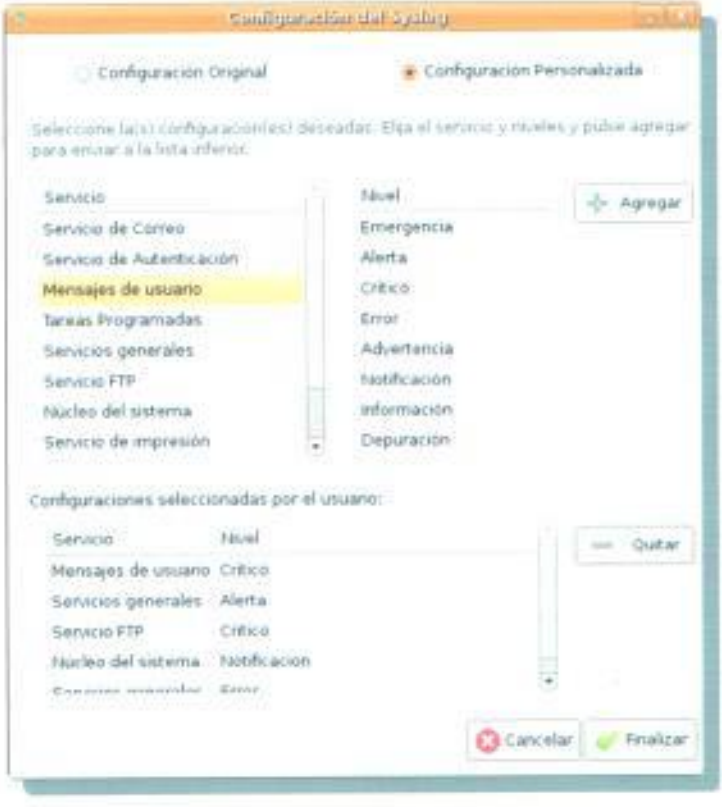

Figura C. 8. Configuración personalizada.

MÓDULO MANTENIMIENTO

El siguiente módulo que encontramos en nuestro sistema es el de "*Mantenimiento*" el cual permite al usuario administrador respaldar, restaurar y eliminar los mensajes de logs según

las necesidades que se requieran. Hay que recordar que la generación de logs a diario puede generar demasiada información dependiendo de la ocurrencia de los eventos en el sistema operativo, por lo que este módulo puede ser muy importante para evitar saturar a la base de datos. Estas opciones pueden ser visualizadas en la figura C.9.

Figura C. 9. Mantenimiento del Sistema.

Respaldar logs

Esta opción permite al administrador respaldar los mensajes de logs a un archivo de respaldo, como lo muestra la figura C.10, se puede filtrar tanto por el servicio como por el nivel, además de filtros por periodos de tiempo, la extensión con la que se guarda este archivo es .bkp.

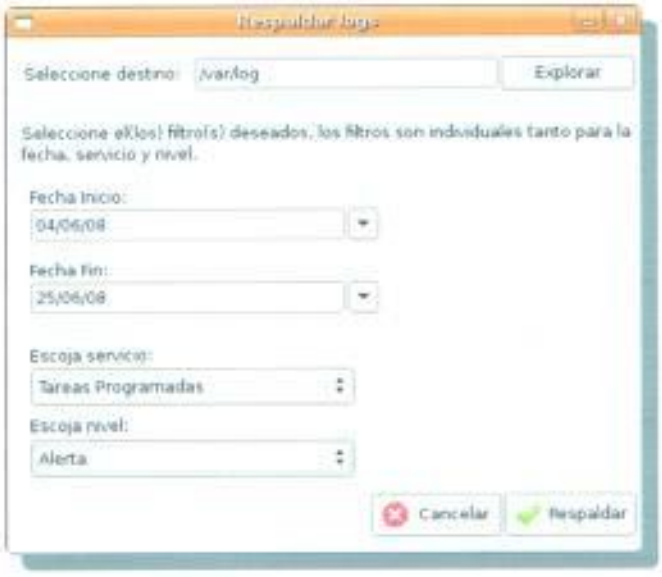

Figura C. 10. Respaldar logs.

Restaurar logs

Si se dispone de algún archivo de respaldo generado previamente, esta opción permite restaurar los mensajes de logs nuevamente a la base de datos, en caso de existir información dentro del respaldo que exista en la base de datos, esta no será restaurada evitando así duplicidad de datos (figura C.11).

Figura C. 11. Restaurar logs.

Eliminar logs

Basándose en filtros por servicio, nivel y rango de tiempo, el administrador puede eliminar mensajes de logs de la base de datos. Es importante mencionar que esta eliminación será permanente por lo cual es siempre necesario y recomendado sacar un respaldo antes de realizar este proceso (figura C.12).

Figura C. 12. Eliminar Logs.

MÓDULO ESTADISTICAS

Este es uno de los módulos más importante del sistema ya que por medio de los gráficos estadísticos se puede analizar el comportamiento de los mensajes de logs en un periodo de tiempo o por su ocurrencia. Se dispone de 3 tipos de gráficos, las líneas de tiempo, histogramas o gráfico de barras y el gráfico circular que han ocurrido. Este módulo se divide en las siguientes opciones tal como se muestra en la figura C.13.

- ✓ Servicio por Incidencia.
- ✓ Prioridad por Incidencia.
- ✓ Servicio por Fecha.
- ✓ Prioridad por Fecha.

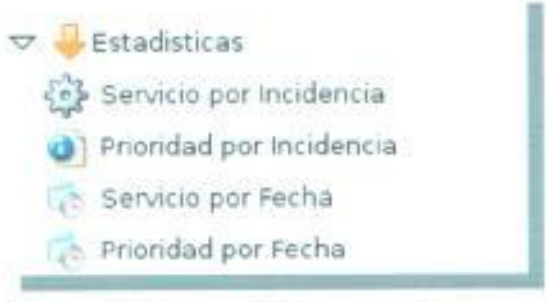

Figura C. 13. Estadísticas del Sistema.

Servicio por incidencia

Esta opción permite generar estadísticas de la cantidad de eventos que ocurren de un servicio en particular y clasificándolos por nivel tal como se puede observar en la figura C.14.

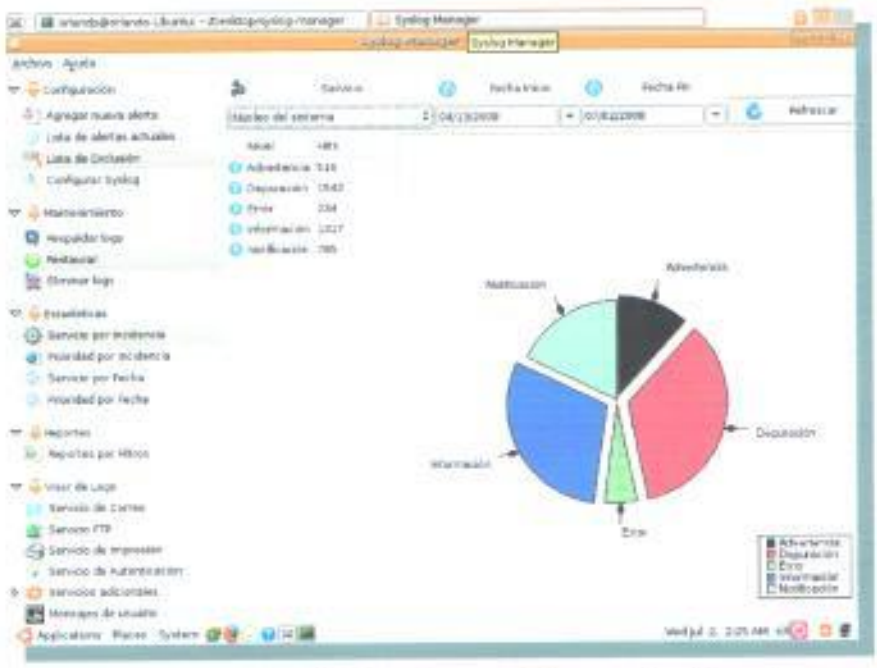

Figura C. 14. Estadísticas de servicios por incidencia.

El usuario administrador adicionalmente tendrá la opción de filtrar estos datos por rangos de tiempo y por servicio.

- **Prioridad por Incidencia**

A diferencia de la primera opción, esta permite la generación de estadísticas que ocurren en general, es decir en todos los servicios, de un nivel o prioridad en particular. Para comprender mejor esto, observe la figura C.15 donde el patrón principal de selección para este caso es el Nivel, y los valores de ocurrencia se muestran clasificados por servicio.

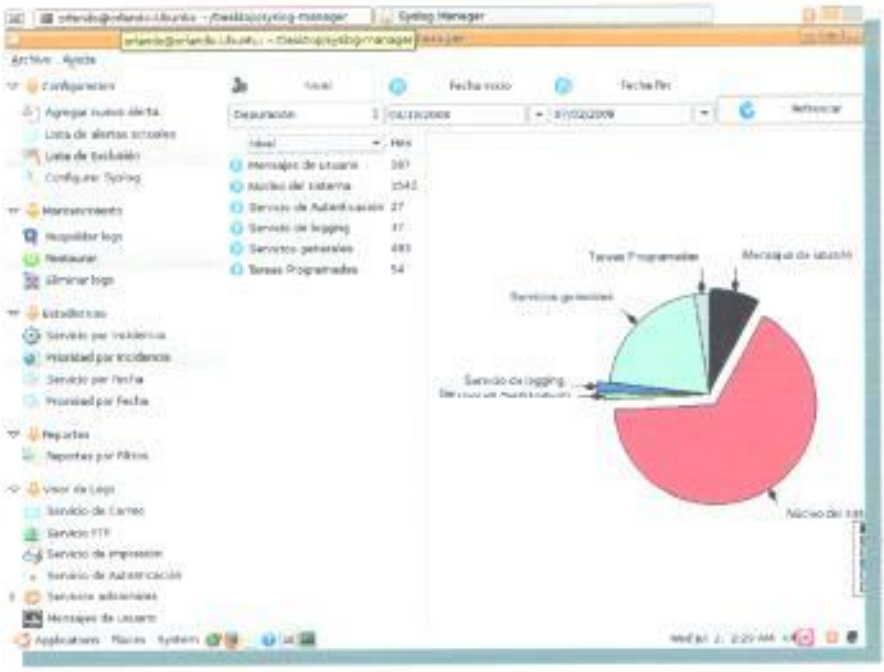

Figura C. 15. Estadísticas de Prioridad por Incidencia.

De igual manera el usuario administrador tendrá la opción de filtrar estos datos por un rango de tiempo y por el nivel.

Servicio Por Fecha

Esta opción muestra diariamente las ocurrencias de los mensajes de logs en líneas de tiempo filtrando por servicio. Por defecto el sistema muestra todos los ocurridos, sin embargo se puede aplicar filtros por rango de tiempo y servicio (figura C.16).

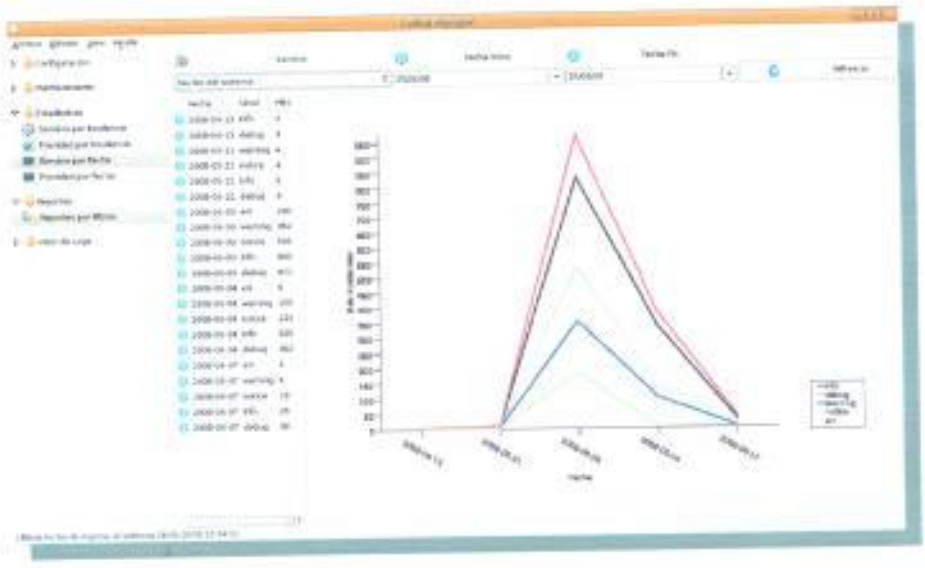

Figura C. 16. Estadísticas de Servicio Por Fecha.

Prioridad por Fecha

Esta opción muestra diariamente las ocurrencias de los mensajes de logs en líneas de tiempo filtrando por nivel. Por defecto el sistema muestra todos los ocurridos, sin embargo se puede aplicar filtros por rango de tiempo y servicio (figura C.17).

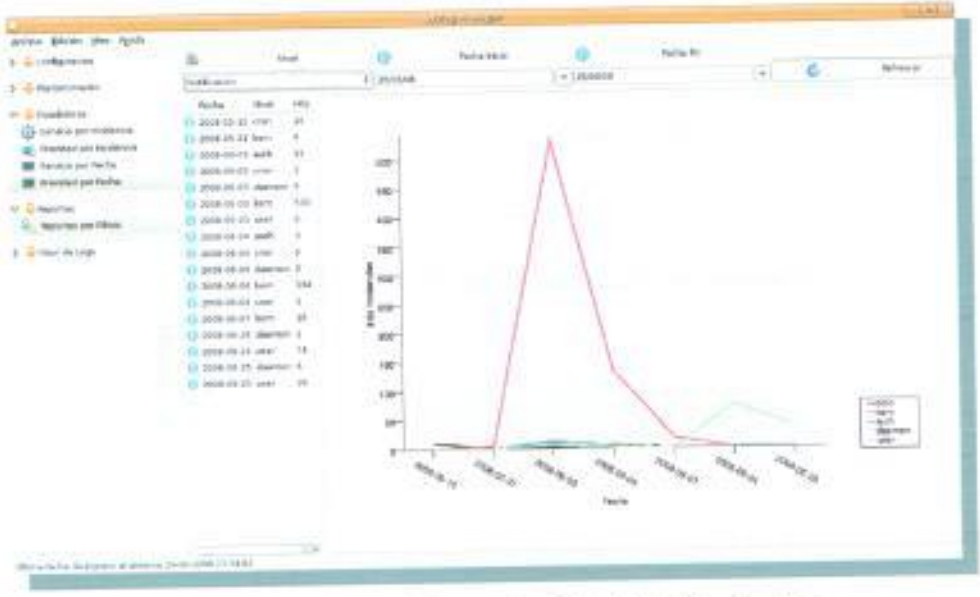

Figura C. 17. Estadísticas de Prioridad Por Fecha.

MÓDULO REPORTES

Permite generar reportes basado en filtros (rango de tiempo, servicio, nivel o palabra a buscar dentro de los mensajes de logs) y exportarlos en formato XML que puede ser visualizado desde un browser o herramientas que manejen el uso de este tipo de archivos (figura C.18).

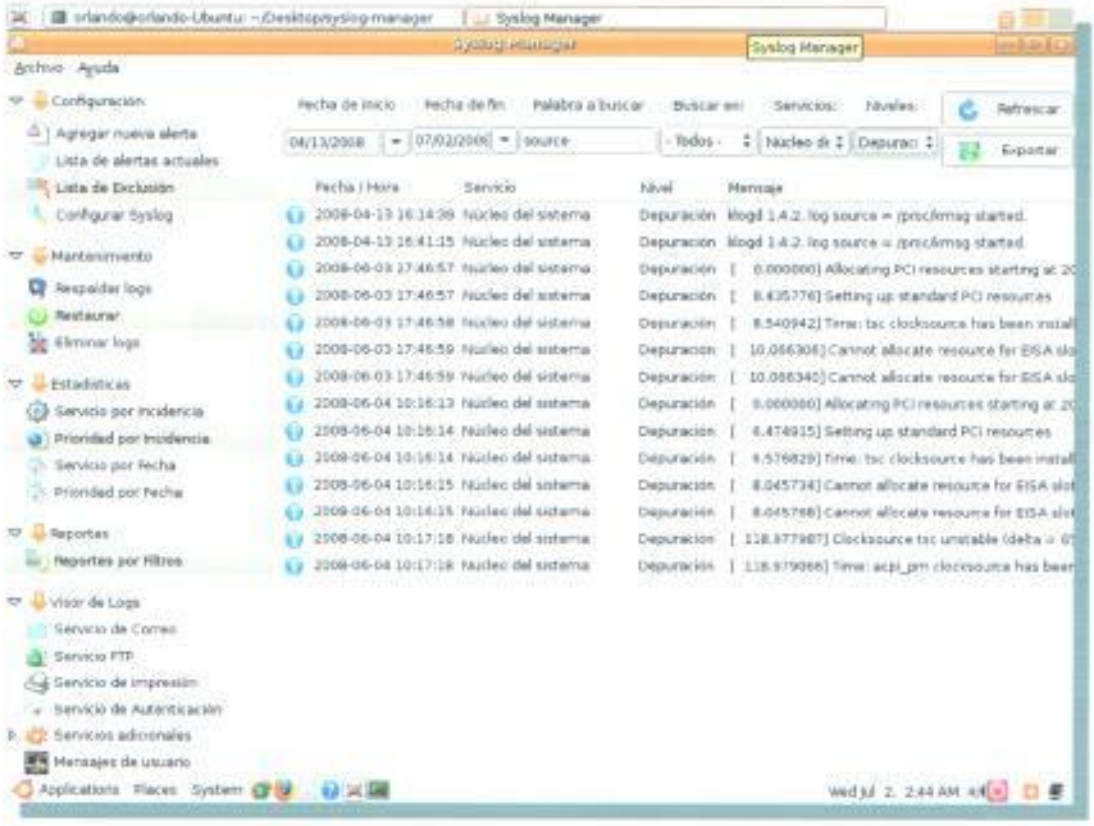

Figura C. 18. Reportes del Sistema.

Una vez filtrado nuestros logs, podemos proceder a generar el reporte en XML por medio del botón "Exportar", para lo cual se nos muestra una ventana, como se visualiza en la figura C.19, donde especificaremos el nombre del archivo y la ruta donde se generará el reporte.

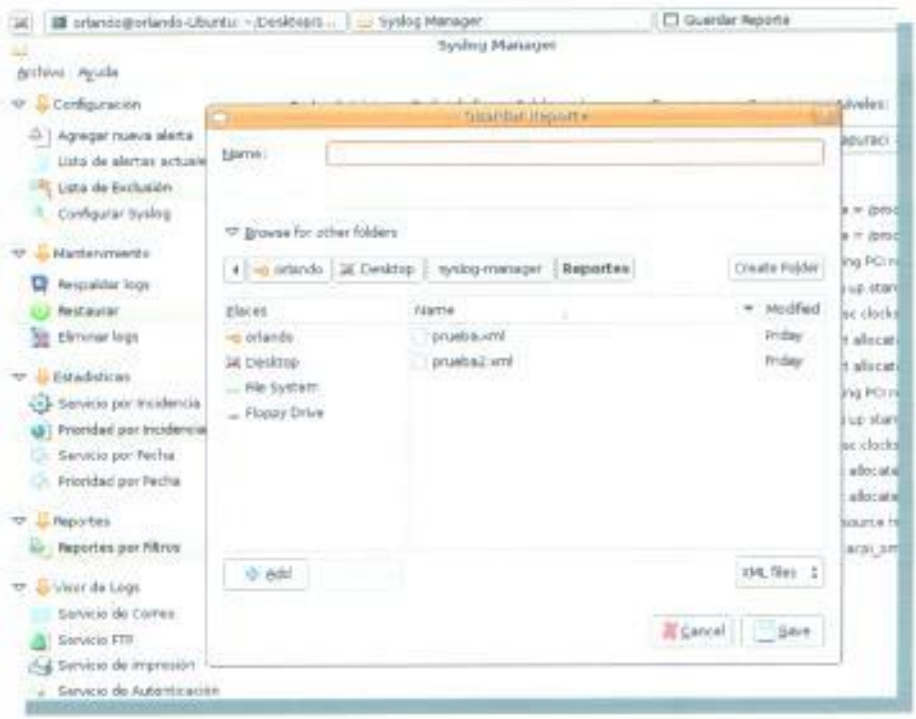

Figura C. 19. Generación en XML de los Reportes.

Es importante conocer y tener en cuenta la ruta donde se guardó el reporte generado en XML, para posteriormente poder cargarlo y visualizarlo en un browser tal como se muestra en la figura C.20.

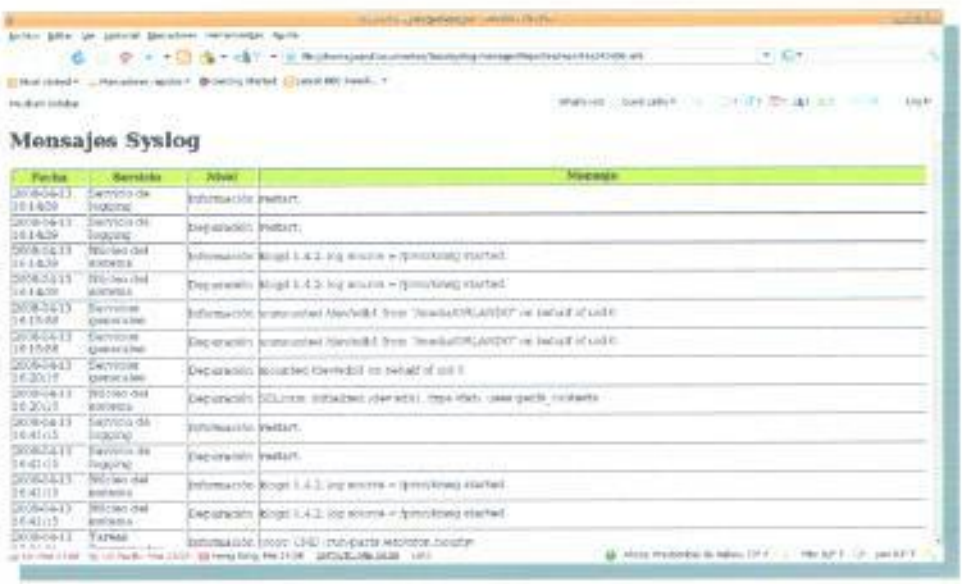

Figura C. 20. Visualización en Web Browser de un Reporte Generado en XML.

MÓDULO VISOR DE LOGS

Finalmente tenemos el módulo "Visor de logs", el cual es el que permite monitorear la aparición de los logs que va generando el sistema. Esta visualización está clasificada por los diferentes servicios que son administrados por el *syslog.conf* tal como podemos observar en la figura C.21.

Figura C. 21. Servicios del Visor de logs.

Como se observa en la figura C.22, el visor de logs permite filtrar por servicio, por rango de fecha, por palabra a buscar dentro de los mensajes de logs y por niveles. La información proporcionada por el visor de logs o eventos suscitados en el sistema está detallada por fecha/hora en que se generó el evento, el demonio que lo generó, el nivel al que pertenece el evento y el mensaje de log generado. Esta información puede ser ordenada seleccionando en la columna que el usuario desee.

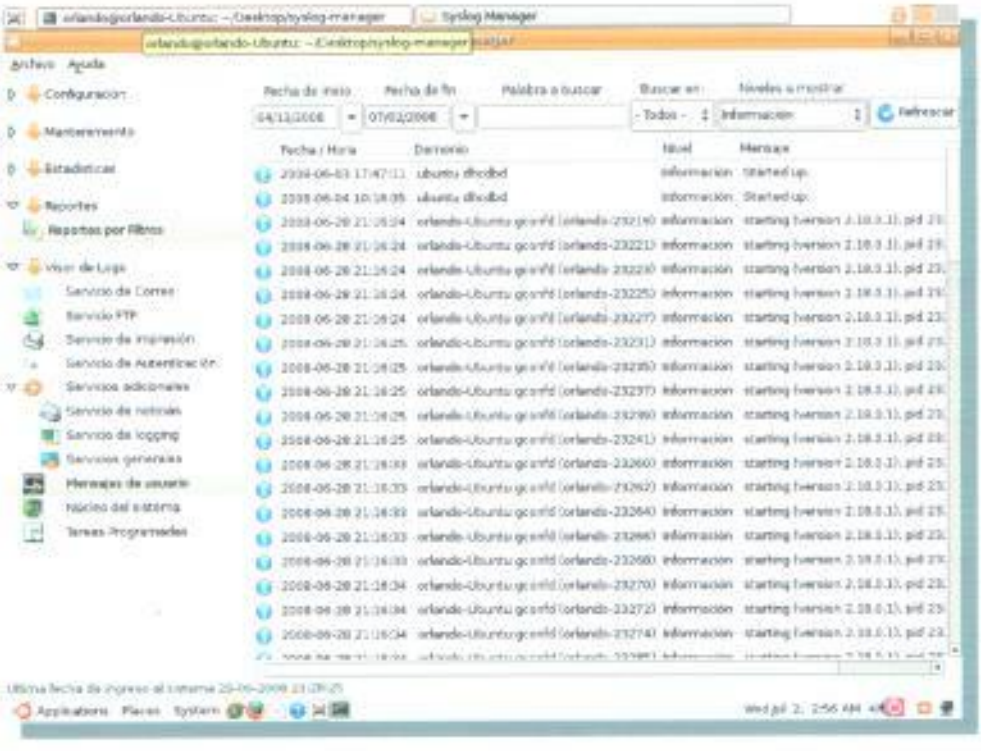

Figura C. 22. Visor de logs del sistema.

Dando clic derecho sobre la lista de mensajes, se puede escoger las siguientes opciones:

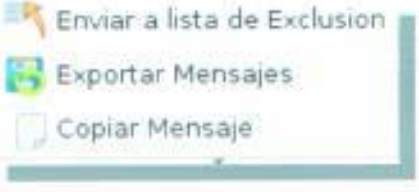

Figura C. 23. Opciones del Visor de logs.

Enviar a lista de exclusión: Permite excluir un mensaje en la visualización de logs, enviando este mensaje a la lista de exclusión.

Exportar mensajes: Permite exportar los mensajes de logs a un archivo XML.

Copiar mensaje: Permite copiar el mensaje de log.