UNIVERSIDAD AUTÓNOMA METROPOLITANA UNIDAD AZCAPOTZALCO

DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA

INGENIERÍA EN COMPUTACIÓN

Documentación

Correlacionador de Bitácoras de equipos en una LAN

Mejía Montes Diana Danae

204359194

Trimestre 10-O

ASESOR.: MARIO ALBERTO LAGOS ACOSTA

No. Económico 22229

PRESENTACIÓN:

El propósito de este documento es de dar a conocer la composición del proyecto terminal “Correlacionador de Bitácoras de equipos en una LAN” destacando la importancia del mismo permitiendo y facilitando la utilización por parte del usuario de dicho Software, garantizando su permanencia para mejoras futuras y así disminuir los costos de operación y de ejecución del proyecto como tal.

AGRADECIMIENTOS

En este proyecto, si bien ha requerido de esfuerzo y mucha dedicación por parte de la autora y su asesor de Proyecto Terminal, no hubiese sido posible su finalización sin la cooperación desinteresada de todas y cada uno de los integrantes de mi familia, amigos, compañeros y personas que involuntaria o voluntariamente participaron, las cuales han sido un soporte muy fuerte en momentos de angustia y desesperación.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS:

INTRODUCCIÓN:

Uno de las principales dificultades que existen en la actualidad, es el tema de la seguridad en las redes de comunicaciones. Las empresas y organizaciones siempre buscan proteger su información y garantizar la integridad, confidencialidad y disponibilidad de sus recursos.

Para lograr esta seguridad, las compañías cuentan con diferentes herramientas de seguridad, tales como firewalls, antivirus, sistemas de detección de intrusos, entre otros. Estas herramientas se encargan de vigilar cierto tipo de información que circula por la red, y en algunos casos permiten alertar al administrador de la red en caso de alguna anomalía.[V]

Es importante tener en cuenta que no es suficiente analizar esta información por independiente. En la mayoría de las veces un simple registro de eventos puede no revelar información importante, pero al relacionarla con registros encontrados en otros puntos de la red o en otros registros del mismo dispositivo, se puede encontrar gran cantidad de información valiosa, sin embargo, teniendo en cuenta el gran número de eventos que ocurren en un tiempo muy corto en una red y la distribución de estos en muchos dispositivos diferentes, resulta muy tedioso y difícil hacer esta relación manualmente.

Hoy en día son complejos los multiniveles de arquitectura de seguridad que consiste de muchos dispositivos para asegurarse que servers, host, y aplicaciones que se ejecutan en la red estén protegidos de actividades maliciosas. Todos estos dispositivos generan voluminosos bitácoras que son difíciles de interpretar dado su consumo e tiempo para analizarlos. Los datos mostrados en dicho bitácoras pueden reflejar actividad normal pero al momento de analizar y correlacionar dichos eventos se puede tener evidencia de eventos anormales, ataques, virus o gusanos.[VI]

Para ello es necesario realizar una correlación de eventos de seguridad. La cual se encarga de encontrar relaciones entre eventos para encontrar información útil. Ésta correlación se logra tras haber obtenido una centralización de los registros también denominados log. [III]

OBJETIVOS DE SISTEMA:

Objetivo General:

Realizar un sistema que proporcione un entorno grafico amigable para el análisis de bitácoras y control de flujo existentes entre comunicaciones en una red de área local para mejorar significativamente la lectura de los registros generados diariamente por los sistemas y servicios.

Objetivos Específicos:

- Investigar y analizar las necesidades de usuarios en el uso de bitácoras.
- Investigar y analizar los registros de bitácoras en diversos sistemas.
- Investigar y analizar las herramientas a utilizar en las múltiples consultas según las entradas.
- Diseñar el sistema correlacionador de Bitácoras que incluya los siguientes módulos.
    1. Diseño e implementación de la Identificación de los diferentes campos en los registros generados.
    2. Diseño e implementación para aislar elementos de búsqueda.
    3. Diseño e implementación para responder en el menor tiempo posible a las consulta realizadas.
- Realizar documentación del correlacionador de Bitácoras.
- Realizar manuales de usuario

DATOS TÉCNICOS

Descripción Técnica:

El sistema correlacionador de bitácoras se plantea como un sistema que mejora considerablemente el análisis de bitácoras en diferentes equipos (figura 1).

En este esquema se da un ejemplo de algunos dispositivos que generan registros incluyendo una LAN

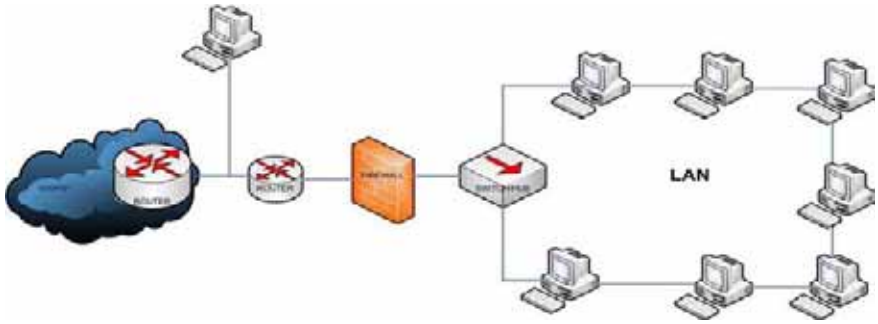

Figura 1:

Especificación Técnica:

Correlación: Se define como un algoritmo que ejecuta una operación de entrada de datos y regresa una salida de datos. El cual trabaja recolectando información y la monitorea de manera parcial, posteriormente este ilumina pequeñas áreas a través del espectro de toda la información que realmente nos interesa. [VIII]

El objetivo del Correlacionador de eventos es:

- Determinar cómo se penetró a un sistema comprometido.
- Determinar qué cambios o modificaciones se cometieron.
- Deslindar responsabilidades a administradores y/o usuarios
- Apoyar a la auditorias
- Reconstrucción de eventos pasados.

Las bitácoras de eventos son un medio para llevar un registro o historial de acontecimientos. Muchos servicios permiten llevar una bitácora de sus actividades.

Para la realización del proyecto depende del protocolo *SYSLOG* que es un estándar de facto para el envío de mensajes de registro en una red informática IP. Por *syslog* se conoce tanto al protocolo de red como a la aplicación o biblioteca que envía los mensajes de registro.

Funciona de la siguiente forma: Existe un ordenador servidor ejecutando el servidor de *syslog*, conocido como *syslogd* (demonio de *syslog*). El cliente envía un pequeño mensaje de texto (de menos de 1024 bytes).

Los mensajes de *syslog* se suelen enviar vía UDP[1], por el puerto 514, en formato de texto plano. Algunas implementaciones del servidor, como *syslog-ng*, permiten usar TCP[2] en vez de UDP, y también ofrecen *Stunnel*[3] para que los datos viajen cifrados mediante SSL/TLS[4].

[1] User Datagram Protocol (UDP) es un protocolo del nivel de transporte basado en el intercambio de datagramas

[2] TCP (Transmission-Control-Protocol, en español Protocolo de Control de Transmisión) es uno de los protocolos fundamentales en Internet. Fue creado entre los años 1973 - 1974 por Vint Cerf y Robert Kahn.

Aunque *syslog* tiene algunos problemas de seguridad, su sencillez ha hecho que muchos dispositivos lo implementen, tanto para enviar como para recibir. Eso hace posible integrar mensajes de varios tipos de sistemas en un solo repositorio central.[I]

El proyecto se considerara como concluido cuando:

1. Se logren establecer una visualización clara en cada una de las consultas de los campo:
   - Fecha
   - Hora
   - IP
   - Tipos de alertas

     Al mismo tiempo que cada pampo se pueda ordenar según su importancia por ejemplo en el caso de fechas se podrá ordenara de mayor a menor valor o viceversa según lo requiera el usuario.
2. Dichas consultas puedan ser exportables en un formato legible en este caso .csv y .txt
3. La interfaz sea amigable e intuitiva para el usuario.

[3] Stunnel es un programa de computadora libre multi-plataforma, utilizado para la creación de túneles TLS/SSL

[4] Secure Sockets Layer -Protocolo de Capa de Conexión Segura- (SSL) y Transport Layer Security -Seguridad de la Capa de Transporte- (TLS), su sucesor, son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

CASOS DE USO:

DIAGRAMAS DE CASO DE USO:

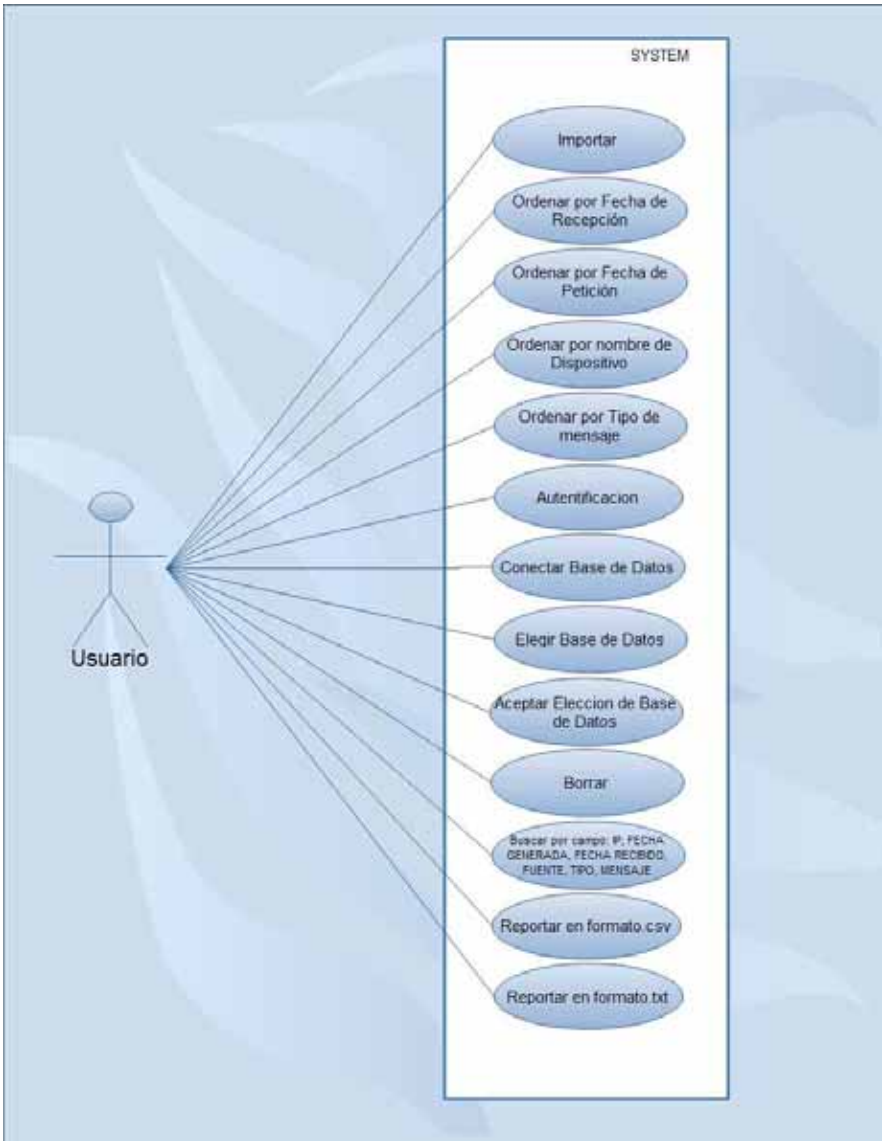

Figura 2: DIAGRAMAS DE CASO DE USO

TABLAS DE CASO DE USO

UC1: Importar

Actor Principal: Usuario
Precondiciones: • **El usuario debe tener acceso al sistema** • **El usuario debe saber el archivo a analizar**
Post-condiciones: • **El usuario pobra abrir cualquier tipo de archivo de texto plano** • **El usuario podrá cancelar la petición**
Escenario Principal: 1. **Sistema despliega Pantalla Inicial.** 2. **Usuario selecciona Importar.** 3. **Sistema despliega la opción Abrir.** 4. **Usuario selecciona la opción Abrir** 5. **Sistema abre la ventana Abrir** 6. **Usuario selecciona archivo a revisar**

Tabla 1: Importar

UC2:Reporta

Actor Principal: Usuario
Precondiciones: • **El usuario debe tener acceso al sistema** • **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos**
Post-condiciones: • **El usuario podrá elegir la opción Ordenar por…** • **El usuario podrá elegir la opción CsV.** • **El usuario podrá elegir la opción TxT.**
Escenario Principal: 1. **El usuario selecciona la opción Reportar** 2. **Sistema despliega las 3 opciones** 3. **Usuario selecciona alguna opción**

Tabla 2:Reportar

UC3: Ordenar por…
Actor Principal: Usuario
Precondiciones: • El usuario debe tener acceso al sistema • Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos • El usuario debe seleccionar la opción Reportar
Post-condiciones: • El usuario podrá elegir la IP • El usuario podrá elegir la FECHA DE PETICIÓN • El usuario podrá elegir la FECHA DE RECEPCIÓN • El usuario podrá elegir el DISPOSITIVO • El usuario podrá elegir el MENSAJE
Escenario Principal: 1. El usuario selecciona la opción Reportar 2. Sistema despliega opciones 3. Usuario selecciona Ordenar por… 4. Sistema despliega Opciones 5. Usuario selecciona alguna opción

Tabla 3: Ordenar por…

UC4: IP
Actor Principal: Usuario
Pre-condiciones: • El usuario debe tener acceso al sistema • Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos • Usuario debe seleccionar opción Reportar > Ordenar por…
Post-condiciones: • El usuario podrá elegir ordena ASCENDENTE • El usuario podrá elegir ordena DESCENDENTE
Escenario Principal 1. El usuario selecciona la opción Reportar 2. Sistema despliega opciones 3. Usuario selecciona Ordenar por… 4. Sistema despliega Opciones 5. Usuario selecciona IP 6. Sistema despliega Opciones 7. Usuario selecciona alguna opción

Tabla 4: IP

UC5: ASCENDENTE (IP)
Actor Principal: Usuario
Precondiciones: • **El Usuario debe tener acceso al sistema** • **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos** • **Usuario debe seleccionar Reportar>Ordenar por…>IP**
Post-condiciones • **Se muestra en la pantalla la información con el orden requerido**
Escenario Principal: 1. **El usuario selecciona la opción Reportar** 2. **Sistema despliega opciones** 3. **Usuario selecciona Ordenar por…** 4. **Sistema despliega Opciones** 5. **Usuario selecciona IP** 6. **Sistema despliega Opciones** 7. **Usuario selecciona Ascendente**

Tabla 5:ASCENDENTE (IP)

UC6: DESCENDENTE (IP)
Actor Principal: Usuario
Precondiciones: • **El Usuario debe tener acceso al sistema** • **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos** • **Usuario debe seleccionar Reportar>Ordenar por…>IP**
Post-condiciones • **Se muestra en la pantalla la información con el orden requerido**
Escenario Principal: 1. **El usuario selecciona la opción Reportar** 2. **Sistema despliega opciones** 3. **Usuario selecciona Ordenar por…** 4. **Sistema despliega Opciones** 5. **Usuario selecciona IP** 6. **Sistema despliega Opciones** 7. **Usuario selecciona Descendente**

Tabla 6: DESCENDENTE (IP)

UC7: FECHA DE PETICIÓN

Actor Principal: Usuario

Pre-condiciones:

- **El usuario debe tener acceso al sistema**
- **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos**
- **Usuario debe seleccionar opción Reportar > Ordenar por…**

Post-condiciones:

- **El usuario podrá elegir ordena ASCENDENTE**
- **El usuario podrá elegir ordena DESCENDENTE**

Escenario Principal

1. **El usuario selecciona la opción Reportar**
2. **Sistema despliega opciones**
3. **Usuario selecciona Ordenar por…**
4. **Sistema despliega Opciones**
5. **Usuario selecciona FECHA DE PETICIÓN**
6. **Sistema despliega Opciones**
7. **Usuario selecciona alguna opción**

Tabla 7: FECHA DE PETICIÓN

UC8: ASCENDENTE (FECHA DE PETICIÓN)

Actor Principal: Usuario

Precondiciones:

- **El Usuario debe tener acceso al sistema**
- **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos**
- **Usuario debe seleccionar Reportar>Ordenar por…>IP**

Post-condiciones

- **Se muestra en la pantalla la información con el orden requerido**

Escenario Principal:

1. **El usuario selecciona la opción Reportar**
2. **Sistema despliega opciones**
3. **Usuario selecciona Ordenar por…**
4. **Sistema despliega Opciones**
5. **Usuario selecciona FECHA DE PETICIÓN**
6. **Sistema despliega Opciones**
7. **Usuario selecciona Ascendente**

Tabla 8: ASCENDENTE (FECHA DE PETICIÓN)

UC9: DESCENDENTE (FECHA DE PETICIÓN)

Actor Principal: Usuario

Precondiciones:

- **El Usuario debe tener acceso al sistema**
- **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos**
- **Usuario debe seleccionar Reportar>Ordenar por…>IP**

Post-condiciones

- **Se muestra en la pantalla la información con el orden requerido**

Escenario Principal:

1. **El usuario selecciona la opción Reportar**
2. **Sistema despliega opciones**
3. **Usuario selecciona Ordenar por…**
4. **Sistema despliega Opciones**
5. **Usuario selecciona FECHA DE PETICIÓN**
6. **Sistema despliega Opciones**
7. **Usuario selecciona Descendente**

Tabla 9: DESCENDENTE (FECHA DE PETICIÓN)

UC10: FECHA DE RECEPCIÓN

Actor Principal: Usuario

Pre-condiciones:

- **El usuario debe tener acceso al sistema**
- **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos**
- **Usuario debe seleccionar opción Reportar > Ordenar por…**

Post-condiciones:

- **El usuario podrá elegir ordena ASCENDENTE**
- **El usuario podrá elegir ordena DESCENDENTE**

Escenario Principal

1. **El usuario selecciona la opción Reportar**
2. **Sistema despliega opciones**
3. **Usuario selecciona Ordenar por…**
4. **Sistema despliega Opciones**
5. **Usuario selecciona FECHA DE PETICIÓN**
6. **Sistema despliega Opciones**
7. **Usuario selecciona alguna opción**

Tabla 10: FECHA DE PETICIÓN

UC11: ASCENDENTE (FECHA DE RECEPCIÓN)
Actor Principal: Usuario
Precondiciones: • El Usuario debe tener acceso al sistema • Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos • Usuario debe seleccionar Reportar>Ordenar por…>IP
Post-condiciones • Se muestra en la pantalla la información con el orden requerido
Escenario Principal: 1. El usuario selecciona la opción Reportar 2. Sistema despliega opciones 3. Usuario selecciona Ordenar por… 4. Sistema despliega Opciones 5. Usuario selecciona FECHA DE RECEPCIÓN 6. Sistema despliega Opciones 7. Usuario selecciona Ascendente

Tabla 11: ASCENDENTE (FECHA DE RECEPCIÓN)

UC12: DESCENDENTE (FECHA DE RECEPCIÓN)
Actor Principal: Usuario
Precondiciones: • El Usuario debe tener acceso al sistema • Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos • Usuario debe seleccionar Reportar>Ordenar por…>IP
Post-condiciones • Se muestra en la pantalla la información con el orden requerido
Escenario Principal: 1. El usuario selecciona la opción Reportar 2. Sistema despliega opciones 3. Usuario selecciona Ordenar por… 4. Sistema despliega Opciones 5. Usuario selecciona FECHA DE RECEPCIÓN 6. Sistema despliega Opciones 7. Usuario selecciona Descendente

Tabla 12: DESCENDENTE (FECHA DE PETICIÓN)

UC13: DISPOSITIVO

Actor Principal: Usuario

Pre-condiciones:

- **El usuario debe tener acceso al sistema**
- **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos**
- **Usuario debe seleccionar opción Reportar > Ordenar por…**

Post-condiciones:

- **El usuario podrá elegir ordena ASCENDENTE**
- **El usuario podrá elegir ordena DESCENDENTE**

Escenario Principal

1. **El usuario selecciona la opción Reportar**
2. **Sistema despliega opciones**
3. **Usuario selecciona Ordenar por…**
4. **Sistema despliega Opciones**
5. **Usuario selecciona DISPOSITIVO**
6. **Sistema despliega Opciones**
7. **Usuario selecciona alguna opción**

Tabla 13: DISPOSITIVO

UC14: ASCENDENTE (DISPOSITIVO)

Actor Principal: Usuario

Precondiciones:

- **El Usuario debe tener acceso al sistema**
- **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos**
- **Usuario debe seleccionar Reportar>Ordenar por…>IP**

Post-condiciones

- **Se muestra en la pantalla la información con el orden requerido**

Escenario Principal:

1. **El usuario selecciona la opción Reportar**
2. **Sistema despliega opciones**
3. **Usuario selecciona Ordenar por…**
4. **Sistema despliega Opciones**
5. **Usuario selecciona DISPOSITIVO Sistema despliega Opciones**
6. **Usuario selecciona Ascendente**

Tabla 14: ASCENDENTE (DISPOSITIVO)

UC15: DESCENDENTE (DISPOSITIVO)
Actor Principal: Usuario
Precondiciones: • El Usuario debe tener acceso al sistema • Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos • Usuario debe seleccionar Reportar>Ordenar por…>IP
Post-condiciones • Se muestra en la pantalla la información con el orden requerido
Escenario Principal: 1. El usuario selecciona la opción Reportar 2. Sistema despliega opciones 3. Usuario selecciona Ordenar por… 4. Sistema despliega Opciones 5. Usuario selecciona FECHA DE PETICIÓN 6. Sistema despliega Opciones 7. Usuario selecciona Descendente

Tabla 15: DESCENDENTE (DISPOSITIVO)

UC16:MENSAJE
Actor Principal: Usuario
Pre-condiciones: • El usuario debe tener acceso al sistema • Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos • Usuario debe seleccionar opción Reportar > Ordenar por…
Post-condiciones: • El usuario podrá elegir ordena ASCENDENTE • El usuario podrá elegir ordena DESCENDENTE
Escenario Principal 1. El usuario selecciona la opción Reportar 2. Sistema despliega opciones 3. Usuario selecciona Ordenar por… 4. Sistema despliega Opciones 5. Usuario selecciona MENSAJE 6. Sistema despliega Opciones 7. Usuario selecciona alguna opción

Tabla 16: MENSAJE

UC17: ASCENDENTE (MENSAJE)

Actor Principal: Usuario

Precondiciones:

- **El Usuario debe tener acceso al sistema**
- **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos**
- **Usuario debe seleccionar Reportar>Ordenar por…>MENSAJE**

Post-condiciones

- **Se muestra en la pantalla la información con el orden requerido**

Escenario Principal:

1. **El usuario selecciona la opción Reportar**
2. **Sistema despliega opciones**
3. **Usuario selecciona Ordenar por…**
4. **Sistema despliega Opciones**
5. **Usuario selecciona MENSAJE**
6. **Sistema despliega Opciones**
7. **Usuario selecciona Ascendente**

Tabla 17: MENSAJE

UC18: DESCENDENTE (MENSAJE)

Actor Principal: Usuario

Precondiciones:

- **El Usuario debe tener acceso al sistema**
- **Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos**
- **Usuario debe seleccionar Reportar>Ordenar por…>MENSAJE**

Post-condiciones

- **Se muestra en la pantalla la información con el orden requerido**

Escenario Principal:

1. **El usuario selecciona la opción Reportar**
2. **Sistema despliega opciones**
3. **Usuario selecciona Ordenar por…**
4. **Sistema despliega Opciones**
5. **Usuario selecciona MENSAJE**
6. **Sistema despliega Opciones**
7. **Usuario selecciona Descendente**

Tabla 18: DESCENDENTE (MENSAJE)

UC19: .CsV
Actor Principal: Usuario
Precondiciones: • El usuario debe tener acceso al sistema • Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos • El usuario debe seleccionar la opción Reportar
Post-condiciones: • El sistema despliega pantalla para guardar • El usuario podrá almacenar el archivo que se encuentra en pantalla en formato .csv • Usuario podrá cancelar la petición
Escenario Principal: 1. El usuario selecciona la opción Reportar 2. Sistema despliega opciones 3. Usuario selecciona .CsV 4. Sistema despliega pantalla para guardar

Tabla 19: CsV

UC20: .TxT
Actor Principal: Usuario
Precondiciones: • El usuario debe tener acceso al sistema • Usuario debe tener archivo abierto el cual se encuentra almacenado en la base de datos • El usuario debe seleccionar la opción Reportar
Post-condiciones: • El sistema despliega pantalla para guardar • El usuario podrá almacenar el archivo que se encuentra en pantalla en formato .csv • Usuario podrá cancelar la petición
Escenario Principal: 1. El usuario selecciona la opción Reportar 2. Sistema despliega opciones 3. Usuario selecciona .TxT 4. Sistema despliega pantalla para guardar

Tabla 20: .TxT

UC21: Ayuda
Actor Principal: Usuario
Precondiciones: • **El usuario debe tener acceso al sistema**
Post-condiciones: • **El sistema despliega opción Acerca de…**
Escenario Principal: 1. **El usuario selecciona la opción Ayuda** 2. **Sistema despliega la opción** 3. **Usuario selecciona la opción**

Tabla 21: Ayuda

UC21: Acerca de…
Actor Principal: Usuario
Precondiciones: • **El usuario debe tener acceso al sistema** • **Usuario debe seleccionar Ayuda**
Post-condiciones: • **El sistema despliega ventana informativa**
Escenario Principal: 4. **El usuario selecciona la opción Ayuda** 5. **Sistema despliega opción** 6. **Usuario selecciona Acerca de…** 7. **Sistema abre ventana informativa**

Tabla 22: Acerca de…

UC22: Autentificación (Conectar)

Actor Principal: Usuario

Precondiciones:

- **El usuario debe tener acceso al sistema**
- **El usuario debe estar registrado en MySQL**
- **El usuario debe de ingresar sus datos**

Post-condiciones:

- **El usuario seleccionara la opción conectar**

Escenario Principal:

1. **Sistema despliega la Pantalla Inicial.**
2. **Usuario ingresa nombre de Usuario**
3. **Usuario Ingresa Contraseña**
4. **Usuario seleccionara CONECTAR**
5. **Sistema verifica información en MySQL**
6. **Sistema despliega BD**

Tabla 23: Autentificación (Conectar)

UC23: Elegir Base de Datos

Actor Principal: Usuario

Precondiciones:

- **El usuario debe tener acceso al sistema**
- **El usuario debe estar registrado en MySQL**
- **Usuario debe ingresar datos**
- **Usuario selecciona conecta**

Post-condiciones:

- **El usuario pobra seleccionar la base de datos**

Escenario Principal:

1. **Sistema despliega la Pantalla Inicial.**
2. **Usuario ingresa nombre de Usuario**
3. **Usuario Ingresa Contraseña**
4. **Sistema verifica información en MySQL**
5. **Sistema BD**
6. **Usuario selecciona BD**
7. **Sistema despliega BD**

Tabla 24: Elegir Base de Datos

UC24:Aceptar

Actor Principal: Usuario

Precondiciones:

- **El usuario debe tener acceso al sistema**
- **El usuario debe estar registrado en MySQL**
- **El usuario debe ingresar datos**
- **El usuario debe selecciona conecta**
- **El usuario debe seleccionar nombre de BD**

Post-condiciones:

- **El usuario pobra abrir cualquier BD según los permisos designados**
- **El usuario podrá analizar la BD abierta**
- **El podrá realizar búsquedas en la BD seleccionada**
- **El usuario podrá modificar la BD**
- **El usuario podrá guardas los cambios en la BD como texto plano**
- **El usuario podrá ordenar por diferentes características la base de datos seleccionada**

Escenario Principal:

1. **Sistema despliega la Pantalla Inicial.**
2. **Usuario ingresa nombre de Usuario**
3. **Usuario Ingresa Contraseña**
4. **Sistema verifica información en MySQL**
5. **Sistema BD**
6. **Usuario selecciona BD**
7. **Sistema despliega BD**
8. **Usuario selecciona BD**
9. **Usuario selecciona Acepta**
10. **Sistema despliega BD**

Tabla 25: Aceptar

UC25:Borrar

Actor Principal: Usuario

Precondiciones:

- **El usuario debe tener acceso al sistema**

Post-condiciones:

- **El usuario pobra borrar cualquier contenido en pantalla**

Escenario Principal:

1. **Sistema tiene desplegado contenido**
2. **Usuario selecciona Borrar**
3. **Sistema elimina cualquier caracter en pantalla**

Tabla 26: Borrar

UC26:Buscar

Actor Principal: Usuario

Precondiciones:

- **El usuario debe tener base de datos desplegada**

Post-condiciones:

- **El usuario pobra buscar información en uno o más campos que son: IP, Fecha generada, Fecha Recibido, Fuente, Tipo y Mensaje**

Escenario Principal:

1. **Sistema tiene desplegado contenido**
2. **Usuario ingresa datos en los campos**
3. **Usuario selecciona buscar**
4. **Sistema muestra coincidencias con los datos ingresados en los campos**

Tabla 27: Buscar

PANTALLAS DE SISTEMA

PANTALLA INICIAL: Muestra la imagen principal del programa

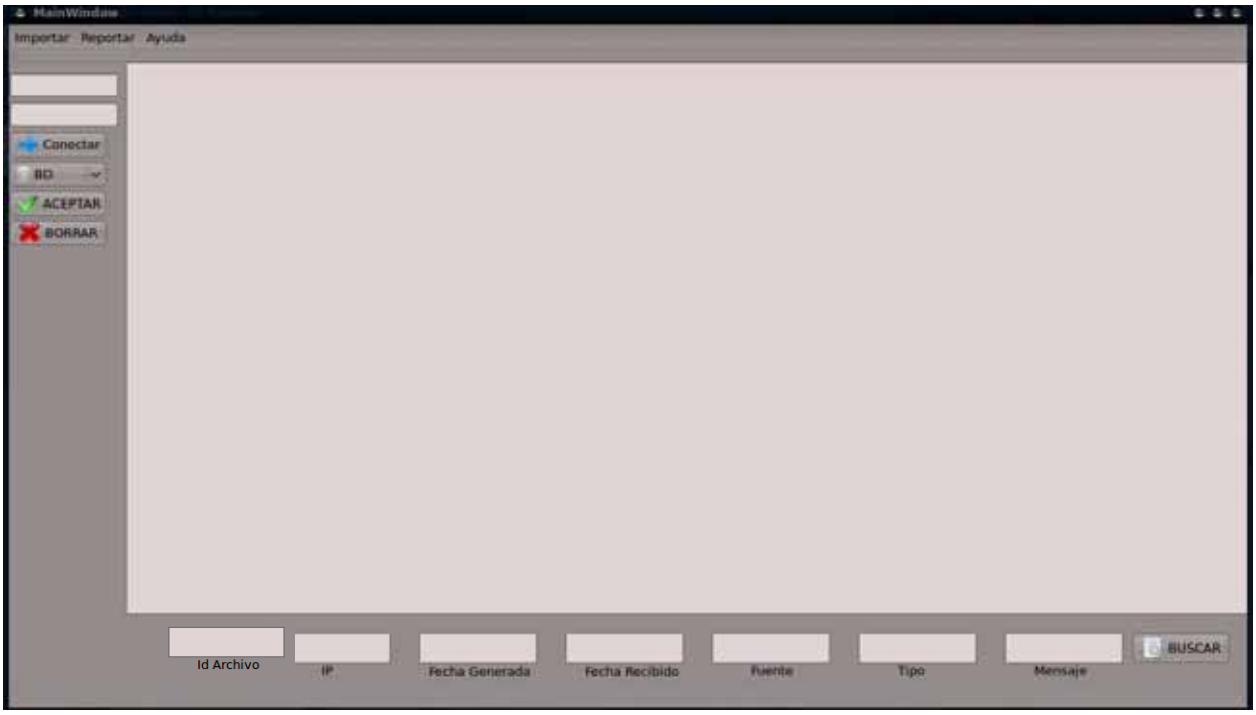

FIGURA 3: Pantalla inicial

PANTALLA 2: Muestra la el Menú Importar, en el cual al seleccionar en el Submenú Abrir se podrán elegir archivos en .txt y .csv únicamente para lectura, estos son archivos que no se encuentran en la base de datos

FIGURA 4: PANTALLA 2

PANTALLA 3: Se ingresa el usuario y la contraseña esto según los privilegios de cada usuario designado previamente el mysql con la finalidad de elegir la base de datos almacenada.

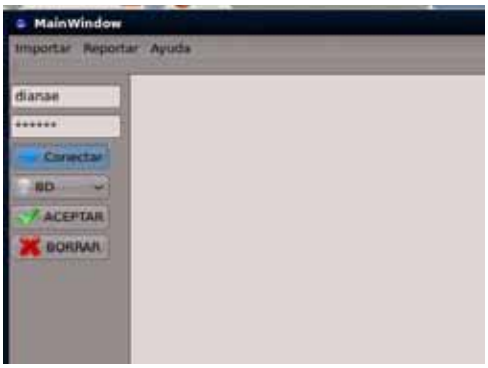

FIGURA 5: PANTALLA 3

PANTALLA 4: una vez conectada muestra ventana de conexión exitosa.

FIGURA 6: PANTALLA 4

PANTALLA 5: una vez realizado el proceso de conexión de muestran las bases de datos a las que el usuario puede tener acceso

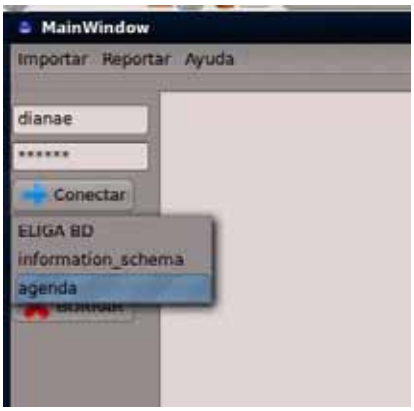

FIGURA 7: PANTALLA 5

PANTALLA 6: una vez elegida la opción se informa al programa el cual se encarga de mostrar los datos almacenados.

FIGURA 8: PANTALLA 6

PANTALLA 7: una vez conectada muestra ventana de conexión exitosa.

FIGURA 9: PANTALLA 7

PANTALLA 8: se muestran los datos almacenados en la base de datos

FIGURA 10: PANTALLA 8

PANTALLA 9: En la parte inferior de la pantalla principal se encuentran campos de escritura, los cuales nos servir al para realizar búsqueda necesaria dentro de la base de datos ya sea con una o más características.

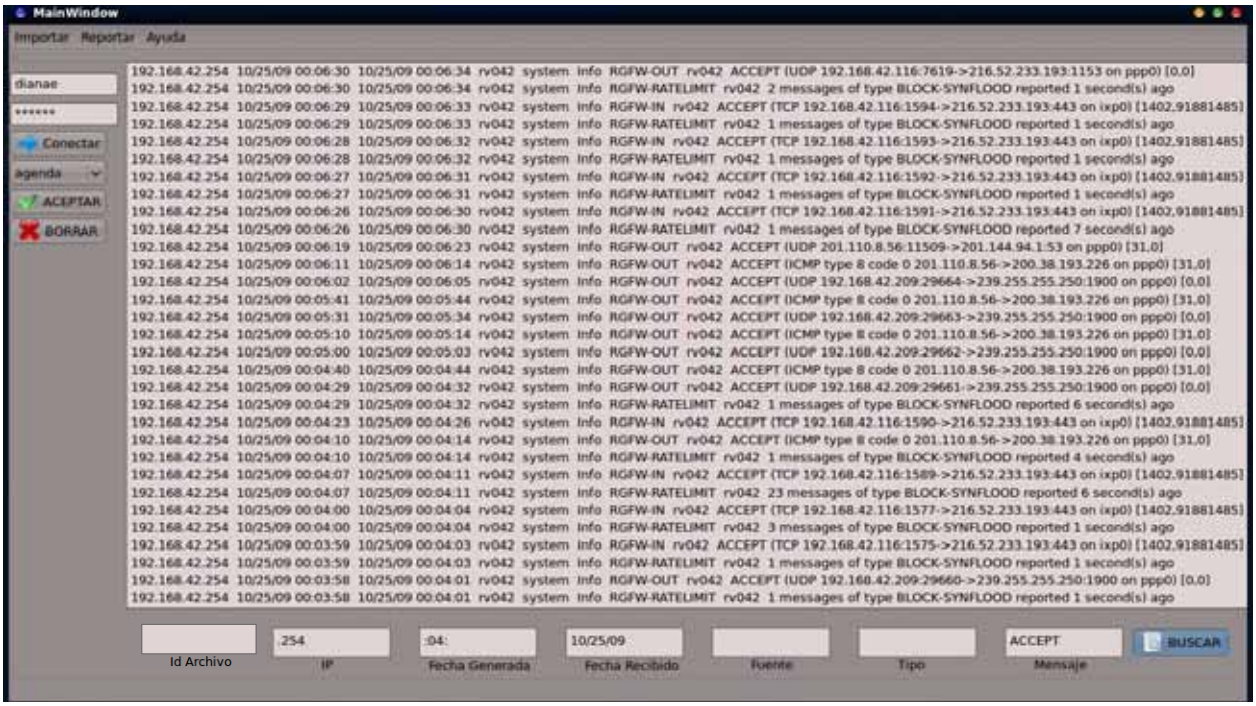

FIGURA 11: PANTALLA 9

PANTALLA 10: Este es un ejemplo

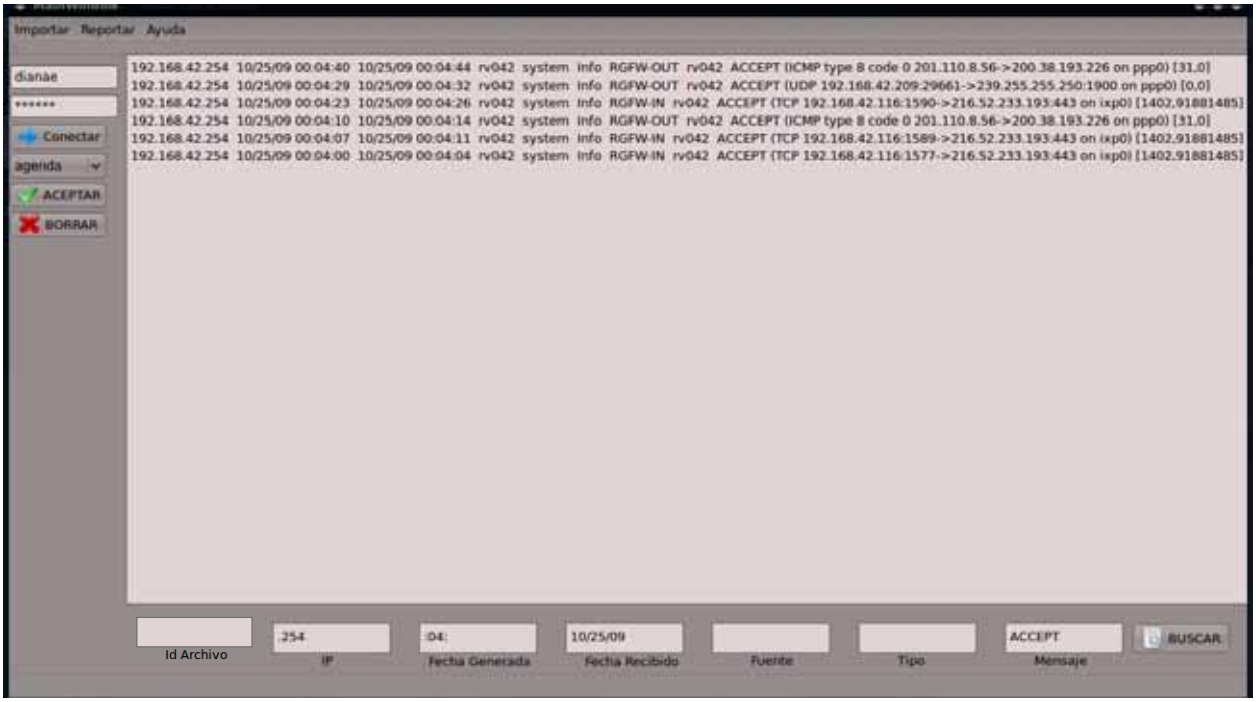

FIGURA 12: PANTALLA 10

PANTALLA 11: Se muestra la opción borrar el cual como su nombre lo indica limpiara todo el contenido que se encuentre en la pantalla.

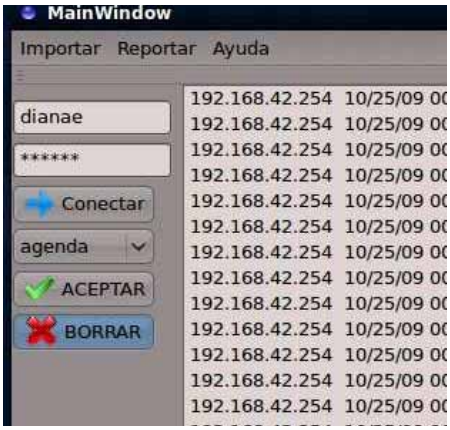

FIGURA 13: PANTALLA 11

PANTALLA 12: Muestra la pantalla limpia, quita el contenido de texto.

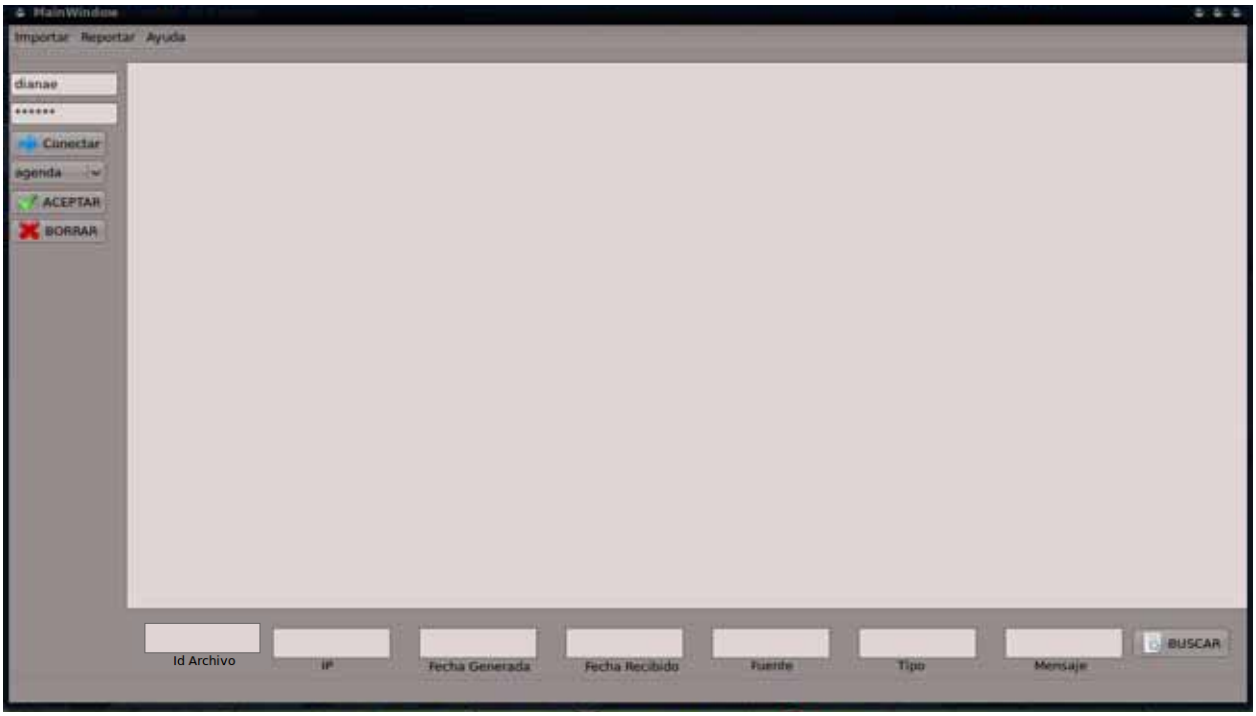

FIGURA 14: PANTALLA 12

PANTALLA 13: Se muestra las diferentes opciones para ordenar la información almacenada en la base de datos la cual puede ser por medio de IP, FECHA DE PETICIÓN, FECHA DE RECEPCIÓN, DISPOSITIVO Y MENSAJE.

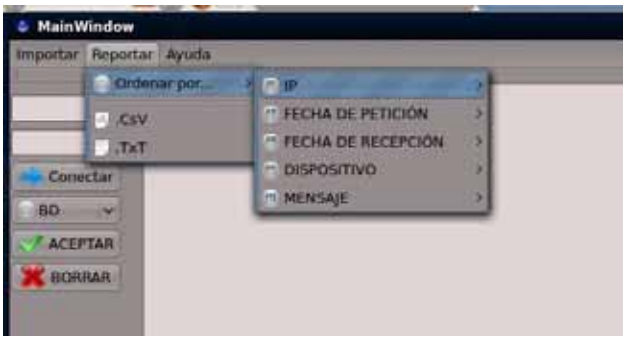

FIGURA 15: PANTALLA 13

PANTALLA 14: una vez seleccionada la caracteristica por la cual vamos a ordenar se elige la opción si se mostrara en orden ascendente o descendente, esta opción se repite en cada elemento.

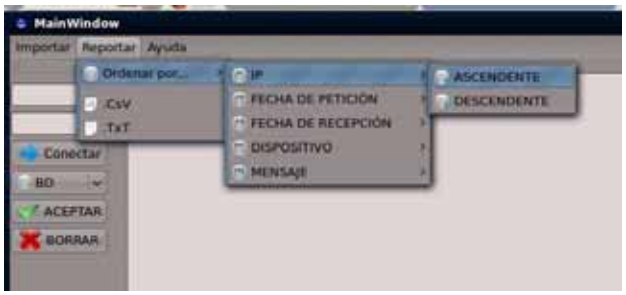

FIGURA 16: PANTALLA 14

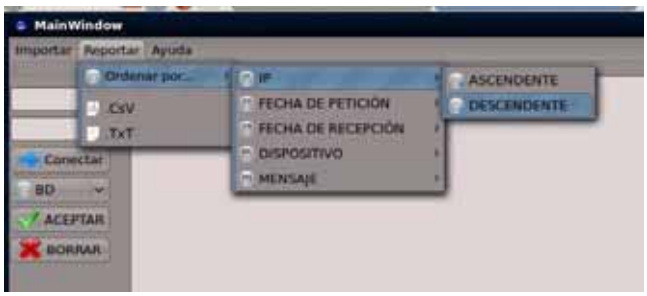

FIGURA 17: PANTALLA 14(1)

PANTALLA 15-16: Se muestra la opción para guardar en formato .csv o .txt

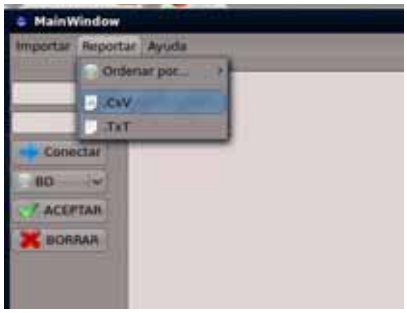

FIGURA 18: PANTALLA 15

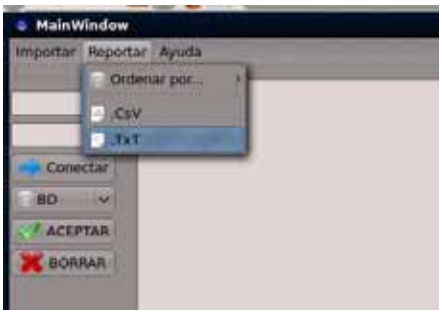

FIGURA 19: PANTALLA 16

PANTALLA 17:

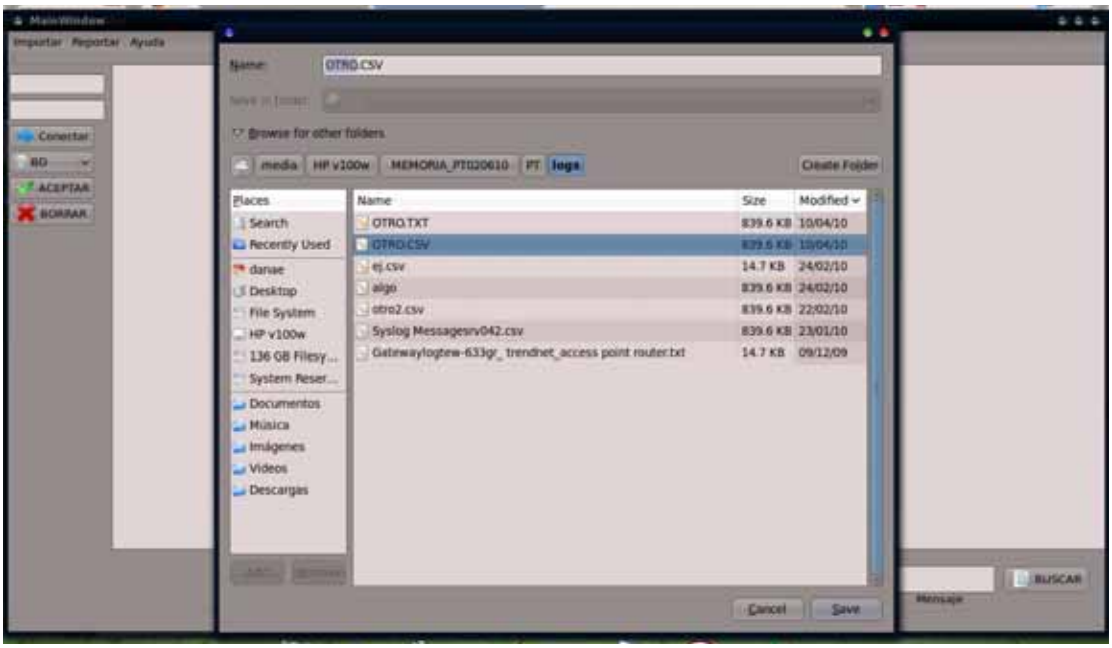

FIGURA 20: PANTALLA 17

PANTALLA 18: En la opción Ayuda despliega Acerca de… el cual despliega una pantalla informativa

FIGURA 21: PANTALLA 18

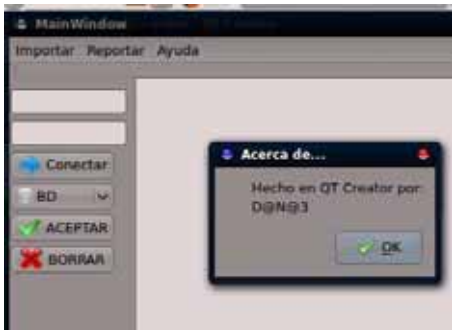

FIGURA 22: PANTALLA 18 (1)

HARDWARE:

El sistema se ha probado en

- Computadora personal 1:

HP Pavilion dv2700 Notebook PC , AMD TURION X2 64 BITS TL 60 2.0 GHz, Nvidia Graphics , Memory Ram 4 GB, System 64 bit, 320 GB Hard Drive, Sistema operativo LinuxMint 6 (felicia), Kernel 2.6.27-15-generic, Sistema de base de datos en MySQL para Linux

- Computadora personal 2:

HP Pavilion t630m, Intel pentium 4 cpu 2.80Ghz, Memory Ram 736 MB, Hard Drive 1: 74,53 GB , Hard Drive 2: 233,76 GB, Sistema operativo: Windows xp con licencia.

SOFTWARE:

El programa se ha probado en SO Linux Mint 6 Felicia y Windows 7

Para crear la interfaz gráfica se realizo sobre Qt4 en sistema operativo Linux Mint 6 Felicia-x64 Edition permitiendo el uso de MySQL para Linux siendo compatible con Windows

Para el manejo grafico de la base de datos se utilizo Navicat 9.0.14

CÓDIGO:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QMessageBox>
#include <qsqldatabase.h>
#include <QSqlError>
#include <QtSql>
#include <QtGui>
////////////////////////////////CONECTANDO BASE DE DATOS///////////////
QString user;
QString login;

MainWindow::MainWindow(QMainWindow *parent)
    : QMainWindow(parent), ui(new Ui::MainWindow)
{
    //QMainWindow l =  new QMainWindow(Login.ui);
    //Ui: login = (new Ui::MainWindow());

    ui->setupUi(this);
}

bool MainWindow::escuchar()
{
QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;
//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
// si logro conectar
QMessageBox::warning(this,"Conection", "Conexion Exitosa");
//consultando
QSqlQuery query;
query.exec("SELECT * FROM base_completa");//te regresa todos

QString salida;
//leyendo salida
while(query.next()){
    // value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"\n";

}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;
```

```
}
MainWindow::~MainWindow()
{
    delete ui;
}
////////////////////////////Abrir archivo/////////////////////////////////
void MainWindow::on_actionAbrir_triggered()
{
   QString fileName = QFileDialog::getOpenFileName(this); /*MUESTRA VENTANA DEL
API DE QT*/
    QFile file(fileName);                                   /*ABRE ATCHIVO Y
GUARDA LA DIRECCION EN UNA CADENA DE TEXTO*/
    if (file.open (QIODevice::ReadOnly|QIODevice::Text))   /*CONVIERTE ARCHIVO
EN CADENA DE TEXTO*/
    {
        ui->plainTextEdit-
>QPlainTextEdit::setPlainText(QString::fromUtf8(file.readAll()));
    }
}
////////////////////////////guardar en
.Csv//////////////////////////////////
void MainWindow::on_action_CsV_triggered()
{
    {
    QString fileName = QFileDialog::getSaveFileName(this);  /*MUESTRA VENTANA
DEL API DE QT*/
    if(fileName.isEmpty())
    return;
    QFile file(fileName + ".csv");
    if(file.open(QIODevice::WriteOnly|QIODevice::Text))
        {
            file.write(ui->plainTextEdit->toPlainText().toUtf8());
        }
    }
}
////////////////////////////////guardar en
.Txt//////////////////////////////
void MainWindow::on_action_TxT_triggered()
{
    {
    QString fileName = QFileDialog::getSaveFileName(this);  /*MUESTRA VENTANA
DEL API DE QT*/
    if(fileName.isEmpty())
    return;
    QFile file(fileName + ".txt");
    if(file.open(QIODevice::WriteOnly|QIODevice::Text))
    {
        file.write(ui->plainTextEdit->toPlainText().toUtf8());
    }
}
}
////////////////////////////////Acerca de...
//////////////////////////////////////////////////
void MainWindow::on_actionAce_rca_de_triggered()
{
         QMessageBox msg;
    msg.about(this,"Acerca de...","Hecho en QT Creator por:\nD@N@3");

}
//////////////////////////////////////////////////////////////////////////////

////////////////////////////////PARA IP ASCENDENTE//////////////////
bool MainWindow::on_actionASCENDENTE_triggered()
{
 QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
```

```
 QString base = ui->comboBox->currentText();

//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);
db.setUserName(user);
db.setPassword(login);
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
// si logro conectar
//QMessageBox::warning(this,"Conection", "Conexion Exitosa");

QSqlQuery query;
query.exec("SELECT Archivos.nombreArchivo, base_completa.* FROM base_completa,
Archivos  where base_completa.idArchivo = Archivos.idArchivo order by IP asc");
QString salida;
//leyendo salida
while(query.next()){
    //value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+" ";
    salida+=query.value(9).toString()+"\n";
}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;

}
////////////////////////////////PARA IP DESCENDENTE//////////////////
bool MainWindow::on_actionDESCENDENTE_triggered()
{
QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;

//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
// si logro conectar
//QMessageBox::warning(this,"Conection", "Conexion Exitosa");

QSqlQuery query;
//query.exec("SELECT * FROM base_completa order by IP desc");
query.exec("SELECT Archivos.nombreArchivo, base_completa.* FROM base_completa,
Archivos  where base_completa.idArchivo = Archivos.idArchivo order by IP
desc");
QString salida;
```

```
//leyendo salida
while(query.next()){
    //value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"  ";
    salida+=query.value(9).toString()+"\n";
}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;
}
////////////para fecha peticion ascendente/////
bool MainWindow::on_actionASCENDENTE_2_triggered()
{
QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;

//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
// si logro conectar
//QMessageBox::warning(this,"Conection", "Conexion Exitosa");

QSqlQuery query;
//query.exec("SELECT * FROM base_completa order by Generated asc");
query.exec("SELECT Archivos.nombreArchivo, base_completa.* FROM base_completa,
Archivos  where base_completa.idArchivo = Archivos.idArchivo order by Generated
asc");
QString salida;
//leyendo salida
while(query.next()){
    //value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"  ";
    salida+=query.value(9).toString()+"\n";

}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;
}
////////////////////para fecha peticion descendente////
```

```
bool MainWindow::on_actionDESCENDENTE_2_triggered()
{
QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;

//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
// si logro conectar
//QMessageBox::warning(this,"Conection", "Conexion Exitosa");

QSqlQuery query;
//query.exec("SELECT * FROM base_completa order by Generated desc");
query.exec("SELECT Archivos.nombreArchivo, base_completa.* FROM base_completa,
Archivos  where base_completa.idArchivo = Archivos.idArchivo order by Generated
desc");
QString salida;
//leyendo salida
while(query.next()){
    //value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"  ";
    salida+=query.value(9).toString()+"\n";

}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;
}
//////////////////fecha de recepcion ascendente//////
bool MainWindow::on_actionASCENDENTE_3_triggered()
{
QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;

//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
// si logro conectar
//QMessageBox::warning(this,"Conection", "Conexion Exitosa");
```

```
QSqlQuery query;
//query.exec("SELECT * FROM base_completa order by Received asc");
query.exec("SELECT Archivos.nombreArchivo, base_completa.* FROM base_completa,
Archivos  where base_completa.idArchivo = Archivos.idArchivo order by Received
asc");
QString salida;
//leyendo salida
while(query.next()){
    //value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"  ";
    salida+=query.value(9).toString()+"\n";
}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;
}
/////////fecha de recepcion decendente/////////////////
bool MainWindow::on_actionDESCENDENTE_3_triggered()
{
QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;

//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
// si logro conectar
//QMessageBox::warning(this,"Conection", "Conexion Exitosa");

QSqlQuery query;
//query.exec("SELECT * FROM base_completa order by Received desc");
query.exec("SELECT Archivos.nombreArchivo, base_completa.* FROM base_completa,
Archivos  where base_completa.idArchivo = Archivos.idArchivo order by Received
desc");
QString salida;
//leyendo salida
while(query.next()){
    //value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"  ";
    salida+=query.value(9).toString()+"\n";
```

```
}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;
}
////////////////////////////// dispositivo ascendente/////////
bool MainWindow::on_actionASCENDENTE_4_triggered()
{
QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;

//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
QSqlQuery query;
//query.exec("SELECT * FROM base_completa order by Source_Name asc");
query.exec("SELECT Archivos.nombreArchivo, base_completa.* FROM base_completa,
Archivos  where base_completa.idArchivo = Archivos.idArchivo order by
Source_Name asc");
QString salida;
//leyendo salida
while(query.next()){
    //value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"  ";
    salida+=query.value(9).toString()+"\n";

}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;
}
////////////////////////////// dispositivo descendente/////////
bool MainWindow::on_actionDESCENDENTE_4_triggered()
{
QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;

//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
```

```
}
// si logro conectar
//QMessageBox::warning(this,"Conection", "Conexion Exitosa");

QSqlQuery query;
//query.exec("SELECT * FROM base_completa order by Source_Name desc");
query.exec("SELECT Archivos.nombreArchivo, base_completa.* FROM base_completa,
Archivos  where base_completa.idArchivo = Archivos.idArchivo order by
Source_Name");
QString salida;
//leyendo salida
while(query.next()){
    //value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"  ";
    salida+=query.value(9).toString()+"\n";

}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;
}

////////////////////////////// mensaje ascendente/////////
bool MainWindow::on_actionASCENDENTE_5_triggered()
{
 QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;

//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
// si logro conectar
//QMessageBox::warning(this,"Conection", "Conexion Exitosa");

QSqlQuery query;
//query.exec("SELECT * FROM base_completa order by Message asc");
query.exec("SELECT Archivos.nombreArchivo, base_completa.* FROM base_completa,
Archivos  where base_completa.idArchivo = Archivos.idArchivo order by Message
asc");
QString salida;
//leyendo salida
while(query.next()){
    //value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
```

```
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"  ";
    salida+=query.value(9).toString()+"\n";

}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;
}
//////////////////////////// mensaje descendente////////
bool MainWindow::on_actionDESCENDENTE_5_triggered()
{
QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;

//datos necesarios para la conexion
db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
// si logro conectar
//QMessageBox::warning(this,"Conection", "Conexion Exitosa");

QSqlQuery query;
//query.exec("SELECT * FROM base_completa order by Message desc");
query.exec("SELECT Archivos.nombreArchivo, base_completa.* FROM base_completa,
Archivos  where base_completa.idArchivo = Archivos.idArchivo order by Message
desc");
QString salida;
//leyendo salida
while(query.next()){
    //value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"  ";
    salida+=query.value(9).toString()+"\n";

}
ui->plainTextEdit->setPlainText(salida);
db.close();
return true;
}
////////////////////////////////////BUSQUEDA////////////////////////////////////
//////////////////////
bool MainWindow::on_pushButton_clicked()

{
    QString entrada_idArchivo = ui->textEdit_id_arch->toPlainText();
    QString entrada_ip = ui->textEdit_ip->toPlainText();
    QString entrada_fechgen = ui->textEdit_fechgen->toPlainText();
```

```
    QString entrada_fechrec = ui->textEdit_fechrec->toPlainText();
    QString entrada_fuen = ui->textEdit_fuen->toPlainText();
    QString entrada_tipo = ui->textEdit_tipo->toPlainText();
    QString entrada_msj = ui->textEdit_msj->toPlainText();


    QString sql = "SELECT Archivos.nombreArchivo, base_completa.* FROM
base_completa, Archivos WHERE IP LIKE '%" + entrada_ip + "%' AND " +
                                    "base_completa.idArchivo IN ( "+
entrada_idArchivo+" ) AND " +
                                    "base_completa.idArchivo =
Archivos.idArchivo AND "+
                                    "Generated LIKE '%" + entrada_fechgen +
"%' AND " +
                                    "Received LIKE '%" + entrada_fechrec + "%'
AND " +
                                    "Source_Name LIKE '%" + entrada_fuen + "%'
AND " +
                                    "Severity LIKE '%" + entrada_tipo + "%'
AND "+
                                    "Message LIKE '%" + entrada_msj + "%'";



QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");
QString base = ui->comboBox->currentText();

QMessageBox msg;

db.setHostName("localhost");
db.setDatabaseName(base);// nombre de base
db.setUserName(user);//usuario
db.setPassword(login);//contrasena

if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return false;
}
//QMessageBox::warning(this,"Conection", "Conexion Exitosa");
QSqlQuery query;
query.exec(sql);

QString salida;
//leyendo salida
while(query.next()){
    // value(0) es el primer campo a leer (columna)
    salida+=query.value(0).toString()+"  ";
    salida+=query.value(1).toString()+"  ";
    salida+=query.value(2).toString()+"  ";
    salida+=query.value(3).toString()+"  ";
    salida+=query.value(4).toString()+"  ";
    salida+=query.value(5).toString()+"  ";
    salida+=query.value(6).toString()+"  ";
    salida+=query.value(7).toString()+"  ";
    salida+=query.value(8).toString()+"  ";
    salida+=query.value(9).toString()+"\n";
}
ui->plainTextEdit->setPlainText(salida);
    query.clear();\
    db.close();
    return true;
}
////////////////////////////////////Conexion con BD
(Autentificacion)////////////
void MainWindow::on_pushButton_5_clicked()
```

```
{

    user = ui->user->text();
    login = ui->login->text();

QSqlDatabase db=QSqlDatabase::addDatabase("QMYSQL");


//datos necesarios para la conexion
db.setHostName("localhost");
//db.setDatabaseName("base_completa");
db.setUserName(user);
db.setPassword(login);
//abrimos la conexion
if(!db.open()){
QMessageBox::critical(0,"Database Error",db.lastError().text());
return;
}
// si logro conectar
QMessageBox::warning(this,"Conection", "Conexion Exitosa");
//consultando
QSqlQuery query;
query.exec("SHOW DATABASES");//te regresa todos
QString salida;
ui->comboBox->clear();
ui->comboBox->addItem("ELIGA BD",1);
while(query.next()){
    // value(0) es el primer campo a leer (columna)
    ui->comboBox->addItem(query.value(0).toString(),1);

}
ui->plainTextEdit->setPlainText(salida);
db.close();

}
```

REFERENCIAS:

[I]. Lorenzo Martínez. Plataformas de consolidación y correlación de eventos [en línea] (2008). [Consulta: 22 Oct. 2009]. <http://www.securitybydefault.com/2008/08/plataformas-de-consolidacin-y.html>

[II]. crash-n-burn. Logs en Linux [en Línea] (2009). [Consulta: 24 Oct. 2009] <http://www.estrellateyarde.es/so/logs-en-linux>

[III]. fwLOGview. Screenshots [en linea] (2009) [Consulta: 30 Oct. 2009] <http://www.nothrix.org/computing/fwlogview/>

[IV]. José Alberto Suárez. Análisis de ficheros log en GNU/Linux [en Línea] (2006) [Consulta: 30 Oct. 2009] <http://www.iberprensa.com/todolinux/articulos/TL65_42-46%20Taller_Log.pdf>

[V]. Grupoica.com. LogICA [en Línea] (2007) [Consulta: 24 Oct. 2009] <http://www.grupoica.com/icaweb/Portals/0/img_webica/logica.pdf>

[VI]. 2MINDS. Servicios Informáticos. Servicios [en línea] (2008) [Consulta: 30 Oct. 2009] <http://www.2minds.com.ar/servicios.swf>

[VII]. LogLady. [en Linea] (2009) [Consulta: 3 Nov 2009] <http://www.kaska.demon.co.uk/loglady.htm>

[VIII]. Eliseo Ortiz Valdez. OSSIM en Fedora Core 4 [pdf] (2009), [Consulta: 23 Oct. 2009]

[IX]. Ramos Alvarez, Benjamin. *Avances en criptologia y seguridad de la información*.[Consulta: permanente]

[X]. Bejtlich, Richard. *El Tao de la monitorización de seguridad en redes* Pearson Educación, s.a., Madrid, 2005. [Consulta: permanente]

Correlacionador de Bitácoras de equipos en una LAN

MANUAL DE USUARIO

DIANA DANAE MEJÍA MONTES

MANUAL DE USUARIO

MANUAL DE USUARIO

SELECCIÓN Importar:

El Usuario inicialmente deberá tener acceso al sistema y tener en mente el archivo que analizara.
En esta opción usuario obra abrir cualquier tipo de archivo de texto plano y también tendrá la opción de cancelar la petición.
En un principio el Sistema despliega Pantalla Inicial, el usuario selecciona Importar y el sistema despliega la opción Abrir, una vez seleccionada el sistema despliega la ventana Abrir y usuario selecciona archivo a revisar.

SELECCIÓN Reportar

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto mismo que se encuentra almacenado en la base de datos.
El usuario podrá elegir la opción Ordenar por…, CsV. y TxT.

SELECCIÓN Ordenar por...

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El usuario al seleccionar la opción Reportar el sistema desplegara diferentes opciones, entre los cuales se encuentran
El usuario selecciona la opción Reportar y Sistema despliega la opción Ordenar por... usuario selecciona la opción y sistema posteriormente despliega las opciones IP, FECHA DE PETICIÓN, FECHA DE RECEPCIÓN, DISPOSITIVO, MENSAJE.

SELECCIÓN IP

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > IP, podrá elegir como ordenar el contenido según la IP registrada, es decir, en orden ASCENDENTE o DESCENDENTE.

SELECCIÓN ASCENDENTE (IP)

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > IP> ASCENDENTE el sistema despliega en la pantalla la información con el orden requerido

SELECCIÓN DESCENDENTE (IP)

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > IP> DESCENDENTE el sistema despliega en la pantalla la información con el orden requerido

SELECCIÓN FECHA DE PETICIÓN

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > FECHA DE PETICIÓN, podrá elegir como ordenar el contenido según la IP registrada, es decir, en orden ASCENDENTE o DESCENDENTE.

SELECCIÓN ASCENDENTE (FECHA DE PETICIÓN)

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > FECHA DE PETICIÓN> ASCENDENTE el sistema despliega en la pantalla la información con el orden requerido

SELECCIÓN DESCENDENTE (FECHA DE PETICIÓN)

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > FECHA DE PETICIÓN > DESCENDENTE el sistema despliega en la pantalla la información con el orden requerido

SELECCIÓN FECHA DE RECEPCIÓN

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > FECHA DE RECEPCIÓN, podrá elegir como ordenar el contenido según la IP registrada, es decir, en orden ASCENDENTE o DESCENDENTE.

SELECCIÓN ASCENDENTE (FECHA DE RECEPCIÓN)

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > FECHA DE RECEPCIÓN> ASCENDENTE el sistema despliega en la pantalla la información con el orden requerido

SELECCIÓN DESCENDENTE (FECHA DE RECEPCIÓN)

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > FECHA DE RECEPCIÓN> DESCENDENTE el sistema despliega en la pantalla la información con el orden requerido

SELECCIÓN DISPOSITIVO

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > DISPOSITIVO, podrá elegir como ordenar el contenido según la IP registrada, es decir, en orden ASCENDENTE o DESCENDENTE.

SELECCIÓN ASCENDENTE (DISPOSITIVO)

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > DISPOSITIVO> ASCENDENTE el sistema despliega en la pantalla la información con el orden requerido

SELECCIÓN DESCENDENTE (DISPOSITIVO)

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > DISPOSITIVO> DESCENDENTE el sistema despliega en la pantalla la información con el orden requerido

SELECCIÓN MENSAJE

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > FECHA DE RECEPCIÓN, podrá elegir como ordenar el contenido según la IP registrada, es decir, en orden ASCENDENTE o DESCENDENTE.

SELECCIÓN ASCENDENTE (MENSAJE)

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > MENSAJE> ASCENDENTE el sistema despliega en la pantalla la información con el orden requerido

SELECCIÓN DESCENDENTE (MENSAJE)

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono las opciones Reportar > Ordenar por… > MENSAJE> DESCENDENTE el sistema despliega en la pantalla la información con el orden requerido

SELECCIÓN .CsV

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono la opción Reportar, el sistema despliega la pantalla para guardar el archivo que se encuentra en pantalla almacenando archivo plano con extensión .csv, si el usuario así lo requiere puede cancelar la petición.

SELECCIÓN .TxT

El Usuario inicialmente deberá tener acceso al sistema y debe tener el archivo abierto, mismo que se encuentra almacenado en la base de datos.
El Usuario una vez que selecciono la opción Reportar, el sistema despliega la pantalla para guardar el archivo que se encuentra en pantalla almacenando en archivo plano con extensión .txt, si el usuario así requiere puede cancelar la petición.

SELECCIÓN AYUDA

El Usuario inicialmente deberá tener acceso al sistema
El usuario al selecciona la opción Ayuda, el sistema despliega la opción Acerca de...

SELECCIÓN Acerca de...

El Usuario inicialmente deberá tener acceso al sistema y al seleccionar Ayuda>Acerca de... el sistema despliega ventana informativa

AUTENTIFICACIÓN ES SISTEMA:

El Sistema para poder Analizar archivos es necesario que estos se encuentren almacenados en una base datos, para tal efecto se utiliza MySQL el cual requiere de usuario y contraseña que serán los mismos datos que proporcionaremos al sistema
El Usuario inicialmente deberá tener acceso al sistema y debe de estar registrado en MySQL para que los datos proporcionados en el sistema coorelacionador coincidan con los que se encuentran en el sistema de gestión de base de datos (MySQL), una ver ingresados los datos el usuario selecciona la opción conectar.

ELEGIR BASE DE DATOS

El Usuario inicialmente deberá tener acceso al sistema, debe de estar registrado en MySQL e ingresar su usuario y contraseña correctamente, posteriormente selecciona la opción conectar, automáticamente el sistema desplegara las bases de datos almacenadas con sus datos.

SELECCIÓN Aceptar

El Usuario inicialmente deberá tener acceso al sistema, debe de estar registrado en MySQL e ingresar su usuario y contraseña correctamente, selecciona la opción conectar y elegir la base de datos, posteriormente seleccionar Aceptar para que el usuario pueda abrir cualquier BD según los permisos designados y poder analizar la BD, realizar búsquedas, modificar la

información de salida(no la que se encuentra en la BD), guardas los cambios en texto plano. También el usuario podrá ordenar por las diferentes características la base de datos seleccionada

Borrar

El Usuario inicialmente deberá tener acceso al sistema para poder borrar cualquier contenido en pantalla, teniendo en cuanta que no se borran los datos en la base de datos, únicamente los que se encuentran en pantalla.

Buscar

El Usuario inicialmente deberá tener acceso al sistema y debe tener base de datos desplegada a la cual analizaremos una vez realizado eso se podrá buscar información en uno o más campos distintos los cuales son: IP, Fecha generada, Fecha Recibido, Fuente, Tipo y Mensaje.