ESCUELA SUPERIOR POLITECNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"Servidor de claves públicas PGP, Cliente Administrador y Cliente para ciframiento y desciframiento de correo electrónico"

Proyecto de Tópico de Graduación previo a la obtención del título de:

INGENIERO EN COMPUTACION

PRESENTADO POR:

Fabián Redrován Castillo

Luis Ruiz Ampuero

Carmen Vaca Ruiz

Ricardo Yánez Godoy

Guayaquil - Ecuador

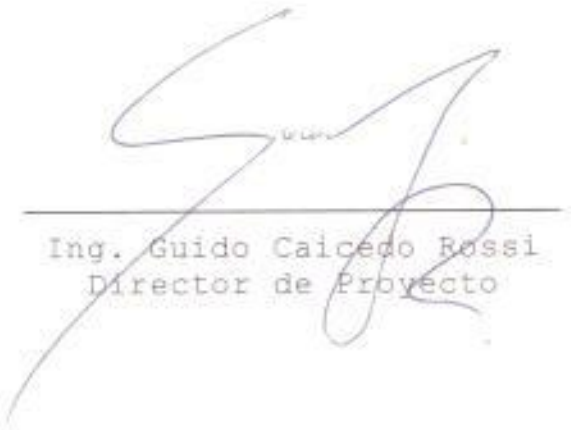
31 de Julio de 1998

A DIOS,

A NUESTROS PADRES,

A NUESTROS HERMANOS,

Y A TODOS AQUELLOS QUE DE ALGUNA U OTRA MANERA PRESTARON SU COLABORACIÓN EN FORMA DESINTERESADA PARA LA CULMINACIÓN DE ESTE PROYECTO.



Ing. Guido Caicedo Rossi
Director de Proyecto

Ing. Carlos Valero
Miembro del Tribunal

Ing. Sergio Flores
Miembro del Tribunal

DECLARACION EXPRESA

La responsabilidad por los hechos, ideas y doctrinas expuestas en este proyecto, excepto en aquellas partes donde se anota explícitamente lo contrario, nos corresponden exclusivamente, y el patrimonio intelectual, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL.

(Reglamento de exámenes y títulos profesionales de la ESPOL).

Fausto Fabián Bedrován Castillo

Luis Medardo Ruiz Ampuero

Carmen Karina Vaca Ruiz

Ricardo Ericst Yánez Godoy

RESUMEN

Se construyó una aplicación (Servidor de claves públicas PGP) que proporciona, a quien lo requiera, la clave pública para encriptación PGP (Pretty Good Privacy)[1] de una persona determinada. Dicha persona debió haber generado previamente su par de claves asimétricas[2] con su propia aplicación PGP (la aplicación solicita un identificador de clave que tiene que ser la dirección de correo electrónico) y, haber proporcionado la pública al administrador del servicio para que la incluya en el Servidor de claves. Ahora bien, con el fin de identificar la clave, es necesario que el dueño de la misma la haya generado utilizando su dirección de correo electrónico.

---

[1] PGP es una aplicación desarrollada por Phill Zimmerman y destinada para la encriptación y desencriptación de texto, que utiliza el paradigma del par de claves asimétricas.

[2] Son claves asimétricas porque se requiere una pública para encriptación y una privada para desencriptación y, trabajan por separado.

También se implementó una aplicación de correo electrónico, cuyo entorno es un browser[1] (Netscape, Internet Explorer u otro), que permite enviar texto encriptado. Para esto hace una búsqueda de la clave pública del destinatario en el Servidor de claves públicas y la utiliza para encriptar el mensaje. De igual forma se puede solicitar en él una búsqueda explícita de alguien en particular y averiguar si posee una clave pública.

De igual forma se puede leer el propio correo con esta aplicación, y, si alguno de los mensajes está encriptado con PGP se lo podrá desencriptar proporcionando la clave secreta del destinatario y la frase de paso de la misma.

Además, para que las claves públicas usadas en la aplicación de correo estén debidamente certificadas, se construyó un cliente que permite la administración de

---

[1] Navegador para Internet, puede ser de tipo gráfico o de tipo texto como "lynx".

las mismas, esto es, añadir, consultar y eliminar. Por tal razón es el administrador del servicio quien debe contactar a los dueños de las claves para corroborarlas.

INDICE GENERAL

CAPITULO 1

GENERALIDADES

1.1 DESCRIPCIÓN GENERAL DEL PROYECTO.

Este proyecto tiene una herramienta que permite enviar y leer correo electrónico encriptado usando el esquema del par de claves asimétricas de encriptación de PGP(Pretty Good Privacy), y otra que maneja información delicada en el Servidor de claves públicas PGP.

Para el proyecto se aprovecha del paradigma CLIENTE-SERVIDOR y se emplea para la comunicación entre las aplicaciones, los beneficios que proporciona la arquitectura TCP/IP, logrando esto mediante la utilización de la interface socket.

El servidor de claves es de tipo stateless, su comunicación es orientada a conexión y presta servicio de manera concurrente.

Los clientes corren en el entorno de un browser, se manejan de manera iterativa y su comunicación es orientada a conexión y transaccional, excepto lo referente a la comunicación con los servidores SMTP y POP3.

1.2 OBJETIVOS.

1. Desarrollar una aplicación en el entorno de un browser, que ponga a disposición de los usuarios de correo electrónico una herramienta sencilla y transparente para cifrar y descifrar el texto de sus mensajes. Esta aplicación se valdrá del paradigma del par de claves asimétricas (claves pública y privada) para el cifrado y descifrado de los mensajes, además de poder ser usado como una aplicación de correo electrónico estándar.

2. Para proveer los beneficios de la encriptación a los usuarios de la aplicación de correo arriba

mencionada, se construirá un servidor que mantendrá un banco de claves públicas. Este servidor almacenará dichas claves en espera que un cliente le solicite una de ellas para poder cifrar (encriptar) el texto de los mensajes antes de enviarlos. Cabe anotar que este servidor solo proveerá de claves públicas certificadas y con formato PGP.

3. Elaborar una herramienta que se ejecute en un browser, para la administración del servidor de claves públicas. Esta aplicación tendrá como fin el ingreso, modificación, eliminación y consulta de claves así como de información adicional de los propietarios de las mismas. El usuario de esta aplicación será el único autorizado para realizar dichos cambios en el servidor de claves.

1.3 JUSTIFICACION DEL PROYECTO.

Siendo el correo electrónico una de las aplicaciones más utilizadas en la Internet, y en vista de que no ofrece una completa seguridad en cuanto a la privacidad de la información que transporta, resultaría de gran

utilidad una herramienta que intente incrementar tal seguridad.

Una manera de suplir ciertas fallas en la privacidad, es el uso del ciframiento del texto de los mensajes previo a su envio.

Para lograr dicho ciframiento, nuestro proyecto se vale de un algoritmo que utiliza un par de claves asimétricas de encriptación. Este procedimiento es implementado por varias herramientas, de las cuales se eligió PGP (Pretty Good Privacy) por ser una de las más difundidas y estar disponible en la Internet para una amplia variedad de plataformas (UNIX®, DOS™, Windows™, OS/2™).

PGP es totalmente gratuito y fácil de utilizar.

1.4 REQUERIMIENTOS FUNCIONALES.

1.4.1 REQUERIMIENTOS PARA EL SERVIDOR PGP.

1. El servidor mantendrá un banco de claves públicas del sistema de encriptación PGP, manejando además datos adicionales asociados con el dueño de esa

clave, como son su nombre, teléfono y dirección de correo electrónico. Todos estos datos son guardados bajo un identificador único que es la dirección de correo electrónico del propietario.

2. Todos los datos serán almacenados en el servidor ya sea usando una base de datos o archivos. Quedando este requerimiento a libre elección de los diseñadores del programa.

1.4.2 REQUERIMIENTOS PARA EL ADMINISTRADOR DE CLAVES PGP.

El cliente administrador de claves PGP podrá realizar transacciones como ingreso, modificación, eliminación y consulta.

Ingreso. Los datos ha ingresarse dependerán de los campos que maneje el servidor de claves pero como mínimo deberán ser la clave pública PGP y el nombre del propietario de dicha clave.

Modificación. Mientras no se definan apropiadamente los campos que manejará el servidor los datos que podrán ser modificados solo podrán ser los datos

adicionales de carácter informativo (es decir, todos menos el correo electrónico del usuario y su clave).

Eliminación. Después de realizar una consulta a la base de datos el dueño de una clave que coincida con el patrón de búsqueda podrá ser eliminado de la base.

Consulta. Una manera de realizar la consulta podrá ser por el correo electrónico del dueño de la clave.

1.4.3 REQUERIMIENTOS PARA EL CLIENTE DE CORREO ELECTRONICO.

El cliente de correo electrónico podrá realizar la transacción de envio y recepción de correo electrónico.

Envío. El usuario tendrá la opción de enviar correo encriptado o no; la aplicación está diseñada para no permitir el envio de correo sin encriptación a no ser que el usuario indique lo contrario. De elegir encriptar, la aplicación podrá ser configurada para que busque la clave pública del destinatario en el servidor y de encontrarla la utilice para encriptar el mensaje.

De igual manera es la búsqueda en el servidor de claves, en caso de haber más de un destinatario.

Una vez encriptado el mensaje, el usuario puede decidir enviarlo (si elige enviar) o volver a editarlo (si elige desencriptar).

1.5 HERRAMIENTAS UTILIZADAS.

La aplicación servidora está desarrollada en lenguaje Java, lo que le proporciona la libertad de poder correr en una plataforma UNIX, Windows 95/NT o Macintosh sin restar ninguna de sus virtudes. Eligiéndose la plataforma UNIX por ser el recurso más a la mano.

El cliente administrador de claves PGP y el cliente de correo electrónico corre bajo Windows 95 o versión superior y utilizarán el entorno de un browser que soporte lenguaje JAVA, preferiblemente Netscape Navigator versión 2.0 en adelante o Internet Explorer 3.0 o superior.

La implementación del cliente administrador de claves PGP y el cliente de correo está desarrollado en lenguaje JAVA™.

Debido a las caracteristicas de JAVA™ se utilizaron varias librerias a disposición pública, resaltando por su importancia las librerias de cryptix mejoradas por Paul Ramsey para funcionar con applets y utilizadas para el ciframiento y desciframiento de mensajes con PGP.

1.6 RESTRICCIONES.

A continuación se describen algunas de las restricciones impuestas para llevar a cabo este proyecto:

El servidor de claves deberá ser desarrollado en una plataforma Windows NT™ o UNIX™.

Como arquitectura de comunicación se deberá usar TCP/IP.

Los clientes deben desarrollarse en lenguaje JAVA™, implementados para ejecutarse en el entorno de un browser.

En particular el cliente de correo electrónico debe utilizar el algoritmo de PGP (Pretty Good Privacy) para

el ciframiento y desciframiento de los mensajes. Por lo tanto, el servidor solo puede almacenar claves públicas de encriptación con formato PGP.

1.7 LIMITACIONES.

Como la mayoria de los browser aún no soportan las últimas versiones de JAVA™, hay que desarrollar los clientes en JAVA™ 1.0, lo cual nos priva de poder utilizar una amplia variedad de código de versiones posteriores que podrian brindar mayor flexibilidad al momento de diseñar las interfaces.

En Internet es muy popular enviar con el correo electrónico archivos de diferentes formatos y tamaños (comúnmente conocidos como "attachments"), facilidad que no posee la aplicación de correo electrónico desarrollada en este proyecto, debido a que JAVA ejecutándose en un browser (aplicación conocida como Applet), no permite ningún tipo de operación con archivos de la máquina que cargó el código JAVA™.

Debido a las restricciones en cuanto al uso de JAVA™, es necesario que el código de los applets (las clases) esté en la misma máquina que el servidor de claves.

El servidor no permite que una clave posea más de un identificador y requiere que el identificador de la misma sea una dirección de correo electrónico, restricciones que PGP no impone sobre las claves. Lo primero es necesario para que la certificación no sea conflictiva, ya que el administrador del servidor de claves públicas PGP sólo puede dar dicha certificación sobre un único identificador y no sobre todos las posibles combinaciones que podrían ser usadas. Lo segundo es para aprovechar un esquema de identificadores únicos y universales ya establecidos, ampliamente conocidos y fáciles de manejar.

1.8 ALCANCE DEL PROYECTO.

Para el futuro se puede lograr un ciframiento de los famosos "attachment", es decir, cifrar gráficos, audio y video.

También se podria leer archivos desde un browser desarrollando el cliente como un plug-in del mismo, o a través del uso de otro lenguaje como ActiveX.

CAPITULO 2

DISEÑO DEL PROTOCOLO

2.1 ARQUITECTURA DE COMUNICACION

El funcionamiento del sistema involucra un flujo de información frecuente entre las aplicaciones que lo conforman, por lo que se hace imprescindible definir los detalles de interacción entre las mismas. Las especificaciones de esta comunicación abarcan tanto un protocolo diseñado especificamente para el proyecto como la utilización de los protocolos SMTP y POPMAIL que son un estándar de Internet.

El Servidor de Claves PGP y los clientes de este proyecto emplean su propia forma de comunicación entre ellos y con el resto de actores involucrados en el sistema, excepto en lo que se refiere a la conexión con los Servidores POPMAIL y SMTP los cuales utilizan para la conexión su propio protocolo de aplicación.

Para obtener una idea global de los requerimientos de comunicación examinemos brevemente el rol de cada una de las aplicaciones que integran el sistema:

a) Servidor de Claves PGP, utiliza una implementación orienta a conexión (ofrece servicio de tipo STREAM), es de tipo stateless (no guarda el estado de los clientes) y concurrente.

Ofrece básicamente dos tipos de servicio:

- Administración de claves públicas, servicio que incluye adición, modificación, eliminación, consulta y validación de claves PGP e información asociada a sus propietarios.

- Transferencia de información entre cliente de correo y servidores SMTP y POPMAIL especificados en la configuración.

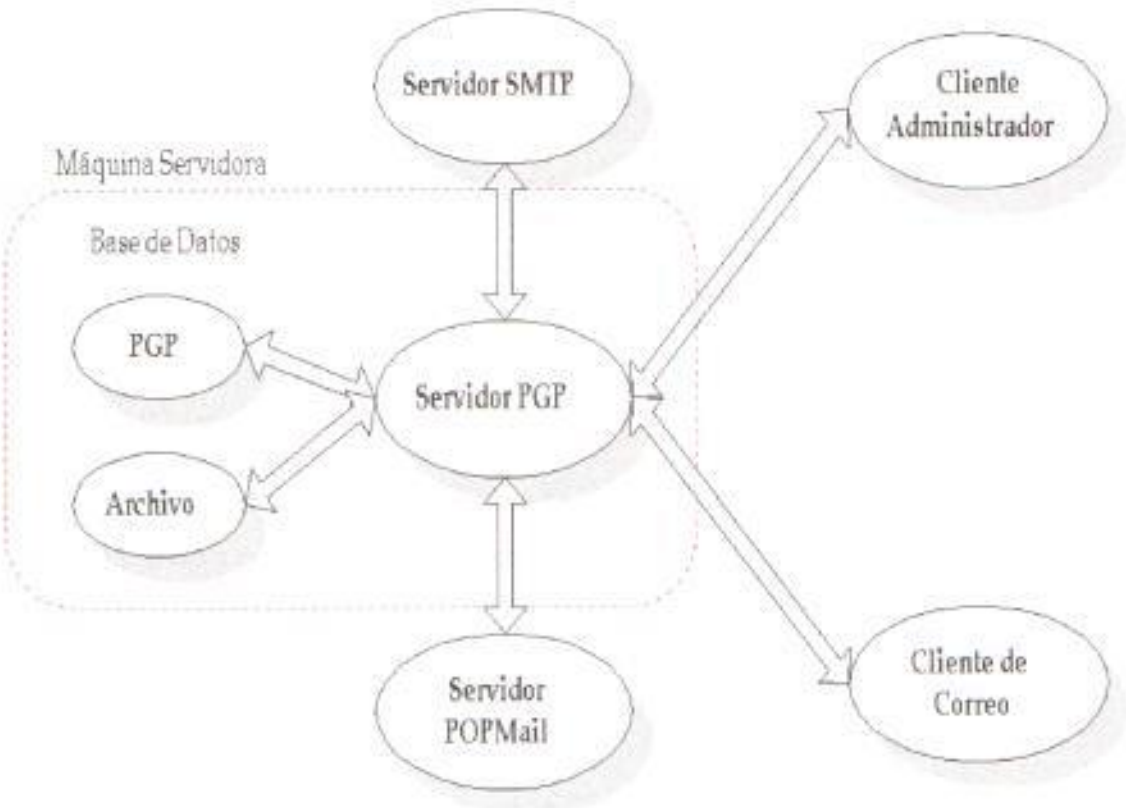


Figura 2.1. Diagrama de interacción Cliente-Servidor

Como muestra el diagrama de la figura 2.1, no tiene ninguna importancia que el Servidor PGP, el Servidor SMTP y el Servidor POPMAIL no residan en una misma máquina, a pesar de que los applets solo puedan

comunicarse con servidores que si lo hagan. Pero si es necesario que la aplicación de encriptación PGP resida en la misma máquina que el Servidor PGP por las restricciones que tiene JAVA al momento de comunicarse con otras aplicaciones. El único actor que podría estar ubicado en una máquina diferente es el archivo de datos, pero la actual versión del sistema por el tipo de archivos que utiliza necesita que éstos residan en la misma máquina.

b) Cliente Administrador, brinda acceso a las opciones de adición, consulta, modificación y eliminación de información del banco de datos que mantiene el servidor de claves.

c) Cliente de Correo, permite enviar o leer correo electrónico usando ciframiento y desciframiento de texto con claves PGP. Este cliente permite además consultar las claves que el sistema tiene disponible.

2.1.1 SERVIDOR PGP VS APLICACIÓN PGP y ARCHIVOS

Ver el esquema en la figura 2.2

Toda la información y manejo de las claves públicas se realiza durante esta interacción. El programa cifrador PGP recibe comandos (ver Apéndice A) proporcionados por el servidor para el manejo de las claves, y éste a su vez envia el resultado de la

Figura 2.2. Servidor PGP vs Aplicación PGP

tarea a un archivo de salida que es manipulado por el servidor.

Si bien es cierto que el servidor trata directamente con PGP para encargarle una tarea, no es así en el otro sentido. PGP envia el resultado de cada una de las tareas ejecutadas a un archivo tipo log que es analizado posteriormente por el servidor a fin de

determinar el status de salida del comando ejecutado. El archivo log es el que permite a PGP comunicarse con el servidor.

Cuando el servidor de claves necesita datos del archivo de información de los propietarios de las claves públicas, no trabaja sólo con el archivo sino también con la aplicación PGP (como se puede apreciar en el diagrama de la figura 2.3), puesto que las claves públicas registradas en el sistema son almacenadas por este programa en un archivo denominado "anillo"

Figura 2.3. Servidor PGP vs Archivo de datos

El anillo de claves públicas no almacena el texto de la clave sino un identificador que le permite generarla en el momento que se la requiera. Bajo este esquema, cuando el servidor necesita sólo una

determinada clave pública y no los datos asociados a ella, no tiene que acceder al archivo de información sino que ejecuta un comando PGP para extraer la clave del anillo.

2.1.2 CLIENTE ADMINISTRADOR VS SERVIDOR PGP

El cliente administrador permite manipular la información que se encuentra en el servidor, para ello posee cuatro opciones Ingresar, Modificar, Eliminar y Consultar. Ver figura 2.4.

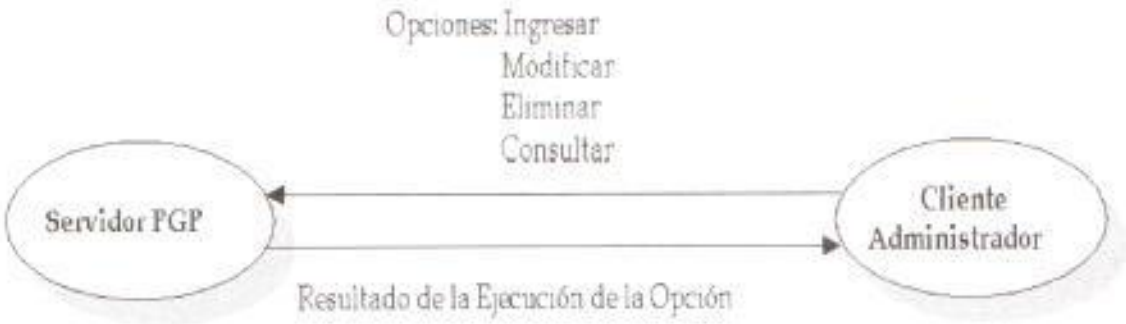


Figura 2.4. Cliente Administrador vs Servidor PGP

En el proceso de autenticación, el cliente administrador envia al servidor un paquete con el user y otro con el password para validar su ingreso al programa.

Una vez que el usuario ingresa a la aplicación el administrador puede añadir nueva información o consultar datos ya existentes, en ambos casos el cliente administrador envía al servidor de claves un paquete que contiene la cadena de caracteres del correo electrónico del nuevo usuario o de uno que ya existe. Dado que este dato se utiliza como identificador de las claves almacenadas en el sistema, es imprescindible suministrar este dato para ejecutar cualquier transacción en el servidor. Junto al identificador, este cliente debe especificar la opción a la que desea acceder.

Cuando se especifica la opción de añadir información nueva, el cliente administrador envía el paquete que posee la información del nuevo usuario, es decir en la parte de datos del paquete TCP va a ir la información del identificador, nombre, apellido, teléfono, y la clave PGP válida.

Cuando se especifica la opción de consultar un usuario que ya existe, el servidor envía al cliente el identificador, nombre y apellido,

teléfono, y la clave PGP. Una vez recibida esta información, el cliente administrador puede:

Modificar: Le envía nuevos datos al servidor.

Eliminar: Le indica al servidor que libere ese registro del archivo de datos.

La opción de consulta permite enviar al servidor PGP patrones de búsqueda.

Un patrón de búsqueda es una cadena de caracteres que puede incluir un máximo de dos caracteres asterisco ('*') como caracteres comodines. En una consulta se localizan todas las coincidencias con los caracteres escritos reemplazando los comodines por cualquier conjunto de cero o más caracteres. Por ejemplo:

"*", "*@*". Localiza todos los identificadores.

"f*". Localiza todos los nombres que empiecen con "f" o "F".

"*@*.espol.edu.ec". Localiza todas las direcciones de correo electrónico que terminen con espol.edu.ec.

Cuando el patrón a consultar se encuentra en el servidor, se devuelve el nombre y el identificador que se estaba buscando. El cliente administrador puede solicitar al servidor PGP información mas detallada (nombre, teléfono y clave PGP).

2.1.3 CLIENTE DE CORREO ELECTRONICO VS SERVIDOR PGP

El esquema de esta comunicación se puede apreciar en el diagrama de la figura 2.5

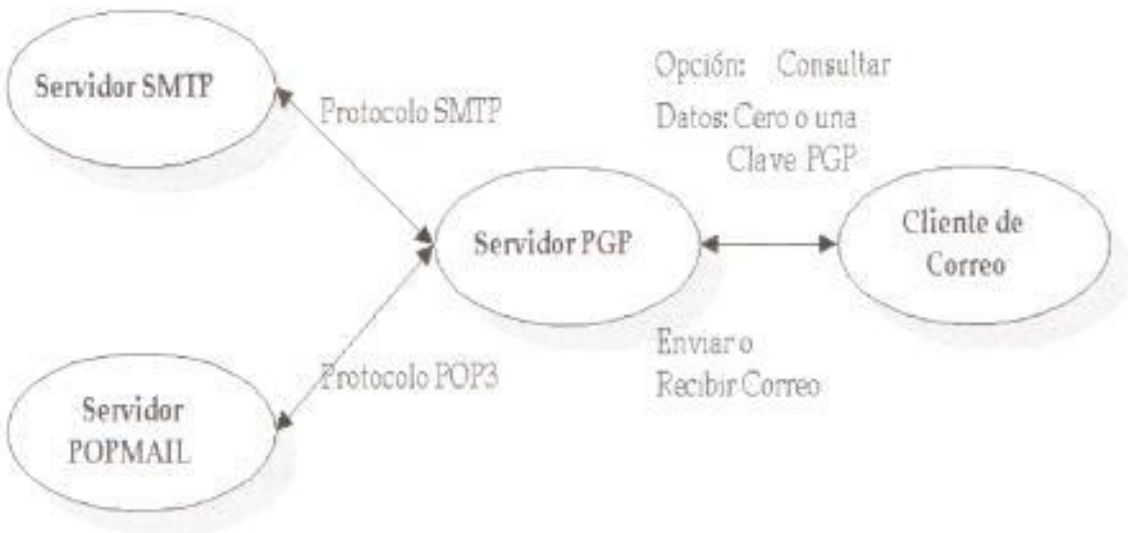


Figura 2.5. Cliente de Correo vs Servidor de claves PGP

El cliente Correo presenta fundamentalmente dos opciones: envio y recepción de correo electrónico. La implementación de cada una de estas opciones incluye el uso de dos estándares previamente definidos en Internet que son SMTP(Simple Mail Transfer Protocol) y POP3 (Post Office Protocol), respectivamente. A más de la utilización de los protocolos mencionados estas opciones implican el hecho de que el cliente de Correo deba interactuar con dos servidores a más del servidor de Claves, el servidor SMTP (hacia el cual enviará correo electrónico) y el servidor POP3 (del cual recibirá correo electrónico).

Para enviar correo electrónico, este cliente se comunica con el servidor SMTP utilizando como "puente" al servidor PGP. La conexión se realiza de la siguiente manera:

Primero el cliente de correo electrónico SMTP envia al servidor de claves una dirección de correo electrónico solicitando una clave PGP, el servidor PGP genera la clave y se la envia al cliente de correo electrónico SMTP, esta conexión se realiza utilizando nuestro propio protocolo.

Una vez que se ha creado la clave para un identificador dado el cliente de correo electrónico SMTP realiza una petición de "puente" al servidor de claves, es aquí donde se establece la conexión con el servidor SMTP utilizando el protocolo SMTP, y entonces, el cliente SMTP puede enviar mensajes hacia el servidor SMTP utilizando la conexión establecida.

De manera similar, para la recepción de correo electrónico este cliente se comunica con el servidor POP3 utilizando como "puente" al servidor de claves. La conexión se realiza de la siguiente manera:

Primero el cliente de correo electrónico POP3 realiza una petición de "puente" al servidor PGP (utilizando nuestro propio protocolo), éste a su vez intenta establecer una conexión con el servidor POP3. Una vez que se establece la conexión - utilizando el protocolo POP3- el cliente de correo electrónico envía el user y password hacia el servidor de claves que enviará esta información hacia el servidor POP3 (también utilizando el protocolo POP3). El servidor POP valida el user y password, envía el resultado de la autenticación al servidor de claves y este a su

vez lo envia al cliente de correo electrónico. Se procede igual con todos los comandos necesarios para "traer" los mensajes de un usuario y luego se cierra la conexión.

El cliente de correo incluye una opción que permite realizar consultas en el servidor de claves de la siguiente manera:

Este cliente envia al servidor PGP un patrón de búsqueda (definido previamente), si el patrón a consultar se encuentra en el servidor, se devuelve el nombre y el identificador que se estaba buscando, pero a diferencia del cliente administrador al seleccionar un item de la lista resultante, éste será incluirlo como destinatario para el envio de correo.

2.2 COMANDOS EN LA COMUNICACIÓN CON SERVIDORES SMTP Y POP3

El cliente de correo electrónico utiliza en la implementación del envio y recepción de mensajes una serie de comandos definidos en el protocolo SMTP y POP3 respectivamente. En esta sección se analiza la secuencia de comandos utilizados para cada opción.

2.2.1 COMANDOS PARA LA CONEXIÓN SMTP

Lo primero que intenta hacer el cliente de correo es conectarse con el servidor de claves y éste conectarse con el servidor SMTP, el cliente espera por un código de estado que refleje que el servidor SMTP ha iniciado la comunicación, este código es el 220 seguido de un espacio en blanco y un mensaje adicional. A continuación se envia un comando que indica si usaremos o no, extensión MIME; para nuestro caso no la usaremos así que el siguiente comando a enviar es HELO y la respuesta que espera el servidor es el código 250 seguida de un mensaje. A continuación usaremos el comando MAIL FROM: <e-mail@host> donde <e-mail@host> es la dirección e-mail de correo electrónico que el usuario colocó como remitente del mensaje, nuevamente el código esperado es el 250.

El siguiente comando es el RCPT TO: <e-mail@host >en donde <e-mail@host > es la dirección e-mail del destinatario del correo electrónico que el usuario introdujo y, se espera por el código 250. Sigue en orden el comando DATA para indicar al servidor el

inicio del cuerpo del mensaje, de aceptarlo el servidor retorna el código 354 seguido de un espacio en blanco y un mensaje. Lo que el cliente hace a continuación es enviar el cuerpo del mensaje linea por línea, finalizando el envio con un carácter de retorno seguido por un carácter de nueva linea, luego un punto y nuevamente un carácter de retorno y uno de nueva linea, el código que se espera es el 250. Por último el comando que el cliente envia es QUIT y el código esperado en el retorno es 21. Es importante anotar que se espera en todos los casos un código respectivo como retorno por parte del servidor SMTP, si dicho código no es retornado, entonces puede algún error en la comunicación con el servidor SMTP, el cliente simplemente detiene el envio y cierra la conexión.

2.2.2 COMANDOS PARA LA CONEXIÓN POP

Lo primero que intenta hacer el cliente de correo es conectarse con el servidor de claves y éste conectarse con el servidor POP, el cliente espera por un código de estado que refleje que el servidor POP3 ha iniciado la comunicación, este código es una

cadena que comienza con "+OK" seguido de un espacio en blanco y un mensaje adicional.

Una vez que la comunicación se ha iniciado exitosamente, el servidor POP3 ingresa al estado de autenticación. Los comandos USER y PASS se usan en combinación para la autenticación, primero el cliente envia el comando USER, si el servidor POP3 responde con un indicador positivo("+OK"), el cliente puede enviar el comando PASS para completar la autenticación o el comando QUIT para terminar la sesión POP3. Si el servidor responde con un indicador de status negativo ("-ERR") al comando USER, el cliente debe intentar nuevamente la autenticación o el comando QUIT.

El servidor puede retornar negativo si existe una cuenta asociada al user enviado, pero el password es incorrecto.

Luego se usa el comando STAT para averiguar el número de mensajes que se habrán de "traer" desde el servidor POP.

Para obtener los mensajes se envía a continuación el comando "RETR" seguido del número del mensaje a traer, el cliente entonces lee las líneas que forman el mensaje hasta encontrar un carácter nueva línea seguido de un punto (".") y otro carácter nueva línea.

Si el usuario especifica que desea borrar su correo del servidor se emplea el comando "DEL" seguido del número del mensaje.

Para terminar la conexión se envía el comando de finalización "QUIT".

Resumen de comandos POP3 usados.

En el estado de Autorización:

USER name

PASS string

QUIT

En el estado Transacción:

STAT

LIST [msg]

RETR msg

DELE msg

QUIT

Respuestas POP3:

+OK

-ERR

2.2.2.1 Descripción detallada de comandos POP3

USER name.

Argumentos

String identificador de un mailbox (requerido), tiene un significado SOLO para el servidor

Restricciones

Solo puede ser usado en el estado Autorización después que el servidor POP3 ha enviado un mensaje indicando que está listo para empezar o después de un comando USER o PASS no exitoso

Posibles Respuestas

+OK name is a valid mailbox

-ERR never heard of mailbox name

Ejemplos

C: USER ryanez

S: -ERR sorry, no mailbox for ryanez here

C: USER mrose

S: +OK mrose is a real hoopy frood

PASS string

Argumentos

Un password de un mailbox especifico del servidor (requerido)

Restricciones

Solo puede ser usado en el estado AUTHORIZATION inmediatamente después de un comando USER exitoso.

Discusión

Después del comando PASS el servidor determina si el usuario tiene acceso al maildrop correspondiente.

Posibles Respuestas

+OK maildrop locked and ready

-ERR invalid password

-ERR unable to lock maildrop

Ejemplo

C: USER lruiz

S: +OK lruiz is a real hoopy frood

C: PASS secret

S: -ERR maildrop already locked

C: USER fredrova

S: +OK fredrova is a real hoopy frood

C: PASS secret

S: +OK fredrova's maildrop has 2 messages (320 octets)

Comando QUIT para el estado AUTHORIZATION:

QUIT

Argumentos

No tiene

Restricciones

No tiene

Posibles Respuestas

+OK

Ejemplos

C: QUIT

S: +OK dewey POP3 server signing off

STAT

Argumentos

No tiene

Restricciones

Solo puede ser usado en el estado Transacción

Posible Respuesta

+OK nn mm,

nn = número de mensajes, mm= tamaño de todos los mensajes juntos.

Ejemplos

C: STAT

S: +OK 2 320

DELE msg

Argumentos

Un número-mensaje (requerido) que NO puede referirse a un mensaje marcado como borrado

Restricciones

Solo puede ser usado en el estado TRANSACTION.

Discusión

El servidor POP3 marca el mensaje como borrado. Cualquier referencia futura a ese número de mensaje asociado con el mensaje en un comando POP3 genera un error. El servidor POP3 no actualiza el borrado del mensaje hasta que la sesión POP3 entra en el estado UPDATE, que ocurre después del comando QUIT.

Posibles Respuestas

+OK message deleted

-ERR no such message

Ejemplos

C: DELE 1

S: +OK message 1 deleted

C: DELE 2

S: -ERR message 2 already deleted

RETR msg

Argumentos

Un número-mensaje (requerido). NO puede referenciar un mensaje marcado como borrado.

Restricciones

Solo puede ser usado en el estado TRANSACTION.

Discusión

Si el servidor POP3 da una respuesta positiva, entonces la respuesta dada es de múltiples líneas. Después que +OK inicia, el servidor POP3 envia el mensaje correspondiente al número de mensaje dato cuidando usar byte-stuff para el carácter de terminación (esto es, si lo encuentra en medio del mensaje, lo manda dos veces para evitar confusiones), como con todos los mensajes de varias líneas.

Posibles Respuestas

+OK message follows

-ERR no such message

Ejemplos

C: RETR 1

S: +OK 120 octets

S: <El servidor POP3 envia el mensaje entero aquí>

S: .

2.3 PROTOCOLO DE APLICACIÓN

El protocolo de aplicación define el formato que tendrán los paquetes de información que se intercambian entre las aplicaciones que integran el sistema. La definición de este protocolo se hace necesario por cuanto el servidor ejecuta transacciones diferentes de acuerdo al requerimiento que recibe. Asi mismo, recibe y envia datos distintos de acuerdo al proceso que realice.

La especificación del protocolo requiere un análisis del servicio a prestar y de los problemas que pueden

surgir en la comunicación. Estos tópicos se analizan en las secciones siguientes.

2.3.1 SERVICIO.

Administrar claves públicas ya sea esto añadirlas al servidor, modificarlas, proveerlas o eliminarlas cuando así lo requiera un usuario debidamente autorizado en caso de ingreso, modificación o eliminación.

2.3.2 PROBLEMA.

El cliente debe especificar en el requerimiento cuál es la opción que solicitará al servidor y de acuerdo a la misma enviar los campos que sean necesarios para que se procese dicha petición. El servidor debe conocer el formato de ese requerimiento.

Según la opción requerida, el servidor enviará una respuesta diferente tanto en cantidad de campos como en longitud de los mismos (estos campos podrían constar de clave pública, datos referentes a usuarios registrados en el sistema, código de respuesta,

etc.). El cliente debe conocer el formato de esa respuesta.

Basándose en estas consideraciones se hace necesario definir un formato y sintaxis para los requerimientos que se enviarán al servidor de claves.

2.3.3 FORMATO Y SINTAXIS DE LOS REQUERIMIENTOS AL SERVIDOR DE CLAVES.

El orden de los campos y el carácter de los mismos dado en el protocolo especifica qué información se debe enviar o leer primero (dependiendo de la aplicación) y qué campos de información son necesarios opcionales (dependiendo de la aplicación y requerimiento o respuesta que se esté enviando).

Se especifica el siguiente orden y opcionalidad para los campos de información:

E-MAIL::ID-OPCION::DATOS OPCIONALES::

Campo E-MAIL: (requerido).

Este el campo que se envía en primer lugar porque es el identificador único de la clave pública utilizado en toda la interacción entre aplicaciones, así que es un dato que se requiere en todo tipo de transacción entre cliente y servidor.

Campo ID-OPCION: (requerido).

Este campo se envía después del E-MAIL porque es el último campo fijo del que dependen los datos que vendrán o no luego.

Campo DATOS OPCIONALES.

Se envían al final siempre que se requieran, de tal manera que no se produzca una ambigüedad cuando no están en el mensaje y la aplicación no confunda datos. Estos pueden ser algunos o todos los siguientes campos:

E-MAIL::ID-OPCION::CLAVE::NOMBRE::TELEFONO::

A continuación, en la Tabla I, se listan los datos que pueden usarse en estos campos dependiendo de la opción utilizada.

TABLA I

OPCION	DESCRIPCION	DATOS OPCIONALES
01	Añadir claves	Clave::nombre::teléfono
02	Modificar datos	Clave::nombre::teléfono
03	Eliminar	
04	Consulta adición	
05	Consulta completa	
06	Consulta completa Admin	
07	Consulta clave	
08	Verificar contraseña	User::passw::
09	Cambiar contraseña	User::passw::newpassw
10	Pedir puente SMTP	
11	Pedir puente POP3	
12	Consulta comodines	
13	Consulta comodines Admin	

2.3.3.1 OPCIONES QUE RECIBE SERVIDOR DE CLAVES

El servidor de claves recibe opciones numéricas, cada una de las cuales representa una tarea distinta a ser ejecutada y se manejan a lo largo de la implementación de clientes y servidores con mnemónicos asociados a cada valor.

Estas opciones que se muestran en la Tabla I, se detallan a continuación:

01 OP_ANAD. Añade una clave nueva e información del propietario: E-MAIL, clave, nombre y teléfono.

02 OP_MODI. Modifica la clave pública y la información del propietario, a excepción del E-MAIL.

03 OP_ELIM. Elimina la clave y la información del dueño que está en el archivo.

04 OP_CONS_EXIS. Hace una consulta del identificador (E-MAIL) y se pide verificar si existe o no.

05 **OP_CONS_TODO.** Hace una consulta del identificador, pero se pide que retorne todos la información que tenga la clave y el dueño.

06 **OP_CONS_TODO_ADMI.** Lo mismo que la anterior, pero realizada por el cliente administrador.

07 **OP_CONS_CLAV.** Hace una consulta del identificador, pero se pide que retorne solamente la clave.

08 **OP_VERI_PASS.** Se solicita verificar que el usuario y contraseña enviados concuerden con los almacenados.

09 **OP_CAMB_PASS.** Además de hacer la misma solicitud que la anterior, se pide cambiar la contraseña con una nueva.

10 **OP_PUEN_SMTP.** Se solicita estar listo para un puente SMTP. En el campo E-MAIL se envia la dirección IP del servidor SMTP.

11 OP_PUEN_POP3. Se solicita estar listo para un puente POPMAIL3. En el campo E-MAIL se envia la dirección IP del servidor POPMAIL3.

12 OP_CONS_WILD. Se solicita realizar una búsqueda con comodines. En el campo E-MAIL se envia el patrón.

13 OP_CONS_WILD_ADMI. Lo mismo que la anterior, pero el requerimiento lo hace el cliente administrador.

2.3.4 FORMATO Y SINTAXIS DE LAS RESPUESTAS DEL SERVIDOR PGP.

Se especifica el siguiente orden y opcionalidad para los campos de información, que se manejarán en las respuestas que el servidor envía a cualquiera de los clientes que ha hecho un requerimiento dado:

ESTADO::DATOS OPCIONALES

Campo ESTADO (requerido).

Este campo tiene la responsabilidad de informar a la aplicación cliente que envió el requerimiento, el resultado obtenido.

Campo DATOS OPCIONALES.

Se envían de ser necesarios y esto depende de la opción que llegó al Servidor PGP en el requerimiento.

A continuación una tabla que lista las posibles respuestas que pueden el servidor de claves puede enviar a los clientes.

TABLA II

ESTADO	DESCRIPCION	DATOS OPCIONALES
01	Adición OK	
02	Modificación OK	
03	Eliminación OK	
04	Identificador existe	Clave::nombre::teléfono
05	Identificador no existe	
06	Identificador múltiple	
07	Clave no válida	e-mail_real::
08	Clave duplicada	e-mail_real::
09	Contraseña OK	
10	Contraseña equivocada	
11	Cambio de contraseña OK	
12	Administrador en linea	
-1	Error desconocido	

2.3.4.1 RESPUESTAS QUE ENVIA EL SERVIDOR DE CLAVES

El servidor de claves envia siempre a los clientes el status de la tarea realizada como un valor numérico, cada una de los cuales representa un status de salida distinto y se manejan a lo largo de la implementación de clientes y servidores con mnemónicos asociados a cada valor.

Estas respuestas que se muestran en la Tabla I, se detallan a continuación:

-1 ES_ERRO_DESC. Retornado cuando se produjo un error desconocido.

01 ES_ANAD_OK. Se retorna cuando la adición de la clave y la información personal fue exitosa.

02 ES_MODI_OK. Se retorna cuando la modificación de una clave y la información existente resultó exitosa.

03 ES_ELIM_OK. Se retorna cuando la clave y la información del propietario fue eliminada exitosamente.

04 ES_IDEN_EXIS. Siempre que exista una clave pública para este identificador, se retorna este estado. Adicionalmente, y, dependiendo del requerimiento, junto a este resultado se puede obtener varios datos opcionales. Asi, por ejemplo, si se solicitó una clave pública el campo que vendrá también será tal clave; ó si se solicitó toda la información referente a un identificador, pues los datos que vienen son la clave, el nombre y el teléfono; ó puede venir toda una lista de nombres, etc. Todo depende del requerimiento.

05 ES_IDEN_NO_EXIS. Se retorna cuando no existe una clave asociada al identificador previamente proporcionado.

06 ES_IDEN_MULT. Se retorna cuando la nueva clave que se pretende ingresar fue creada con más de un identificador.

07 ES_CLAV_MAL. Se retorna cuando la nueva clave que se pretende ingresar está mal formada, dañada o no es un clave PGP. También se retorna cuando la nueva clave que se quiere ingresar fue creada con

un identificador diferente del que se indicó como tal. Solamente en este último caso se retorna otro campo con el identificador (correo electrónico) del verdadero dueño de la clave.

08 ES_CLAV_DUPL. Se retorna cuando la nueva clave que se quiere ingresar fue añadida previamente a nuestros archivos con un identificador diferente del que posee en estos momentos. Esto se puede dar debido a que nuestro identificador es el correo electrónico del dueño, y éste es asociado con un número id que es realmente el que maneja PGP. Además de esto, retorna el identificador (correo electrónico) con el que consta la clave en el servidor.

09 ES_PASS_OK. Se retorna cuando el usuario y la contraseña que empleó el administrador para ingresar son correctos.

10 ES_PASS_MAL. Se retorna cuando el usuario o la contraseña que empleó el administrador para ingresar son incorrectos.

11 ES_CAMB_PASS_OK. Se retorna cuando el cambio de contraseña solicitado por el administrador se realizó exitosamente.

12 ES_ADMI_IN_LINE. Se retorna cuando, el cliente administrador envía algún requerimiento al servidor PGP existiendo ya otro cliente administrador actualmente trabajando con el servidor.

2.3.5 DIAGRAMAS DE ESTADO

Una manera de apreciar el protocolo usado para la comunicación Cliente - Servidor es observar los diagramas de estado tanto de los clientes como del servidor. Estos se incluyen a continuación:

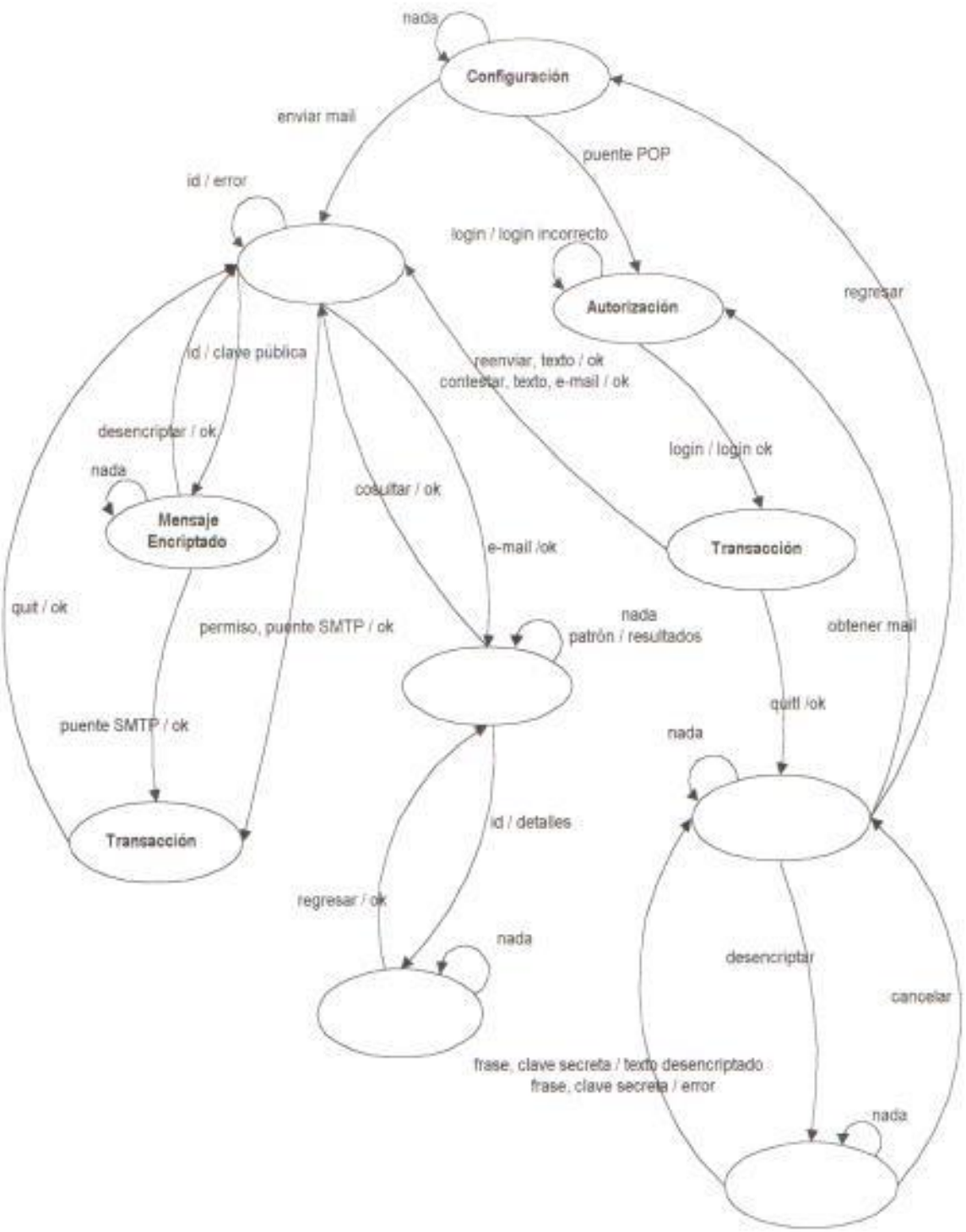


Figura 2.3. Diagrama de Estados del Cliente de Correo Electrónico

CAPITULO 3

DISEÑO DEL SERVIDOR DE CLAVES PGP

3.1 FUNCIONALIDAD DEL SERVIDOR DE CLAVES PGP

Para levantar el servidor de claves debe ubicarse fuera de la carpeta que contiene los archivos con extensión ".class" que forman el servidor (a la fecha de impresión de este documento dicho directorio es ServidorPGP) y tipearse la siguiente línea de comando:

java ServidorPGP.startup -[debug] [elPuerto]

java es el programa interpretador de bycodes (archivos ".class", llamados también simplemente clases) cuyo nombre proviene del lenguaje de programación JAVA que

es el que produce estos bycodes. ServidorPGP es el directorio que contiene la clase que inicia el servidor (startup.class) y deber ser separado de dicho clase con un punto. El modificador "-debug" puede escribirse con mayúsculas o minúsculas indiferentemente e indica al servidor que cualquier anomalia u error grabe o no, sea notificado por pantalla. La acción por defecto especifica no notificar nada por pantalla a excepción del mensaje que notifica que el servidor de claves se ha levantado con éxito o en caso contrario el por qué no pudo iniciarce correctamente. Esta opción es opcional al igual que el modificador "elPuerto", el cual especifica como su nombre lo dice el puerto en que corre el servidor, que es por defecto el 6001.

Por el servicio que ofrecerá, el servidor se lo puede dividir en dos partes. Por un lado tendrá un banco de claves públicas de encriptación (ciframiento) a las cuales permitirá su correspondiente mantenimiento y consulta. Y por otro lado hará de un puente para la aplicación de correo electrónico, para que esta pueda utilizar un servidor SMTP o POP3 en una máquina cuya ubicación en la red sea arbitraria.

El servidor de claves mantendrá comunicación con dos clientes. La primera de ellas será un cliente de correo electrónico, la segunda un cliente que realizará la administración de estas claves con las tareas que ello conlleva.

Los servidores SMTP y POP3 mantendrán una comunicación con la aplicación de correo electrónico y no propiamente con el servidor de claves, es decir la presencia del servidor de claves será más bien transparente desde el punto de vista del cliente de correo y los dos servidores de correo.

El servidor posee un formato propio para los requerimientos y para las respuestas desde y hacia los clientes respectivamente. Estos requerimientos y respuestas no tienen una longitud fija y de tener varias partes utiliza separadores entre las partes, estos requerimientos siempre contienen un campo llamado opción el cual determina el servicio que solicita el cliente. La respuesta de manera paralela al requerimiento, posee siempre un campo de estado que refleja el resultado de la ejecución de la tarea solicitada al servidor.

3.1.1 ¿CÓMO INTERACTÚA CON EL ADMINISTRADOR?

El servidor obliga al cliente administrador a logonearse antes de permitirle realizar cualquier tipo de requerimiento, pues todas aquellas operaciones que pueden llegar a solicitar el administrador se encuentran restringidas a ser solicitadas por un solo administrador a la vez, esto es cuanto un administrador sé logonee no lo podrá hacer otro, hasta que el primero permanezca sin realizar transacción alguna por un lapso mayor a cinco minutos.

El servidor para el logoneo, recibe un requerimiento que consta de un user acompañado de una opción que indica la intención de logonearse y finalmente el password. Estos campos se encuentran divididos por separadores estando el user y password no como texto sino como un número entero. De ser estos datos debidamente verificados por el servidor, este retorna un estado de logoneo correcto o no.

Una vez logoneado el administrador puede hacer operaciones de Ingreso, Consulta de todos los datos,

Eliminación, Consulta de existencia(si esta existe) de una clave y modificación de los datos. Todos estos requerimientos siempre están precedidos por una dirección de correo electrónico que es el único identificador de claves que el servidor de claves reconoce como válido.

A continuación se da una da una descripción de los requerimientos que puede enviar el cliente administrador al servidor de claves.

Al recibir la opción de Ingresar, esta trae consigo a continuación un campo que contiene la clave de encriptación seguida de otros dos campos que contienen el nombre y el teléfono del propietario de dicha clave. El servidor de claves se encarga de verificar primeramente si la clave posee un solo identificador (el servidor de claves rechaza el ingreso de una clave que tenga más de un identificador), luego verifica si este identificador corresponde con la dirección de correo electrónica que llegó con el requerimiento y finalmente si una clave con esta dirección de correo electrónico ha sido ingresado previamente. Si todas las

validaciones anteriores son sorteadas con éxito el servidor de claves le dará al programa PGP la clave de encriptación para que este la guarde en el anillo de claves públicas (archivo usado por PGP para almacenar claves). Seguidamente si PGP guarda la clave con éxito se abrirá el archivo de datos generales para grabar el nombre y el número de teléfono del propietario de la clave. Finalmente el servidor responderá al cliente con un código de estado de ingreso con éxito o de haber algún error en cualquier lugar del proceso se cancelará el ingreso, se reportará el error por pantalla (si el servidor fue arrancado con el modificador -debug) y se retornará al cliente un código tratando de describir, de la mejor manera posible, por qué se canceló el ingreso.

Al recibir la opción de consultar existencia el Servidor de claves se encargará de verificar si existe un identificador registrado que corresponda con esa dirección de correo electrónico. Al cliente se le enviará un código de estado describiendo el resultado de la búsqueda (existe o no).

La opción consultar todos los datos funciona de manera similar a la anterior, diferenciándose solo que al responder se envía el estado de existencia acompañado de la clave que es recuperada por PGP del anillo de claves, el nombre y finalmente el teléfono asociado a la clave recuperados estos dos últimos del archivo de datos generales. De no existir se retorna el estado de no-existencia seguido de campos con el valor de null.

De ser Eliminar la opción recibida, el servidor de claves borra los datos que corresponden a esa dirección de correo electrónica. Esto se logra suministrando a PGP el comando para que elimine la clave pública correspondiente. Luego se borra del archivo de datos generales la información adicional a esta clave. Al final se envía al cliente un código de estado de éxito o fracaso en la eliminación.

La opción de modificar es una combinación de las operaciones elementales (Ingresar, Consultar y Eliminar). Se Comienza salvando los datos viejos (Consultar), luego se remueven estos datos (Eliminar) y finalmente se ingresa la nueva clave

pública y datos asociados a esta. De algo salir mal el servidor restaurará los datos anteriores y retornará un código reflejando el fracaso en la modificación o por el contrario un código que reflejará un éxito en la modificación.

3.1.2 ¿CÓMO INTERACTÚA CON EL CLIENTE DE CORREO ELECTRÓNICO?

El cliente en primera instancia envia la dirección de correo electrónico con una opción que identifica la operación solicitada. Dicha opción puede ser el consultar todo, para lo cual el servidor solicitará a PGP por una clave pública cuyo identificador sea la dirección de correo electrónico que llegó en el requerimiento. En caso de no encontrar PGP la clave en el anillo notificará al servidor de claves el cual a su vez enviará al cliente un código de estado que indique que no se encontró. En caso contrario además del código de estado que indica que el identificador solicitado si existe, se enviará la clave pública y los datos adicionales asociados a esta.

Otra opción permitida para este cliente es la de consulta con wildcardss en donde el servidor busca en el archivo de datos generales por coincidencias con el patrón enviado. Este tipo de búsqueda puede hacerse por correo electrónico si el patrón posee un "@" o por defecto se hace por el nombre de la persona. La respuesta ha esta opción igualmente lleva su código indicando un éxito o fracaso y al ser éxito se incluirá una lista de todas aquellos coincidencias separadas una de otra por caracteres de nueva línea. Por último el tercer tipo de búsqueda es el de solo la clave, donde la respuesta es ahora solo la clave pública de haberla recuperado PGP del anillo o solo el código de fracaso de no encontrarla.

El cliente de correo tiene dos opciones especiales más que, por la manera como funcionan, le da una característica algo singular al servidor. La opción de puente SMTP o POP3 viene junto con la dirección IP o nombre del servidor SMTP o puente POP3 al que se quiere enlazar el cliente de correo. De allí en adelante el servidor actúa de manera transparente

sirviendo solo de intermediario entre el cliente y el respectivo servidor de correo con el que se abrió un puente.

3.2 TIPO DE SERVIDOR Y SU JUSTIFICACION

La concurrencia en un servidor hoy en dia es algo bastante común, pues los usuarios de cualquier red por más modesta que esta sea, exigen rapidez a la hora de navegar en ella a la caza de información o de herramientas que le brinden alguna utilidad. Rapidez que no seria posible ofrecer de elegir una implementación iterativa, pues el procesamiento de todos los servicios que ofrece el servidor de claves requieren de una mayor o menor cantidad de accesos a archivos e incluso comunicación con otro programa (aplicación PGP) lo cual implica un cierto grado de tiempo de procesamiento. Dando el valor que se merecen todas estas realidades la implementación de este servidor es concurrente.

El servidor se basa en transacciones, es decir ante un requerimiento ofrece una respuesta y no tiene sentido

mantener información sobre el estado actual de algún cliente, es decir es un servidor del tipo stateless.

La parte más importante y delicada del servidor es el manejo que le da a las claves públicas, ya que un solo carácter fuera de posición ocasionaria que la clave quedara inservible, esto fue razón suficiente para elegir TCP como la capa de transporte en vez de UDP, otorgándole a nuestro servidor la característica de orientado a conexión, con todos los beneficios que esto trae como control de errores, retransmisiones, manejo de paquetes perdidos, duplicados o en desorden.

3.3 DATOS MANEJADOS POR EL SERVIDOR

La razón por la que se incluye esta sección es porque el servidor de claves mantiene sus datos de una manera poco usual. Para entender esto es necesario saber que el servidor hace uso de dos elementos importantes, para el manejo de las claves públicas y la información asociada a estas. El primer elemento importante que maneja el servidor es un archivo que almacena información personal de los propietarios de las claves, como son su nombre, dirección de correo electrónico y

teléfono. Los datos son divididos en el archivo por separadores y la información de cada usuario ocupa una sola línea. Este archivo es de solo texto y se guarda bajo el mismo directorio donde están las clases del servidor.

El segundo elemento importante que opera con el servidor, es la aplicación PGP, cuyo trabajo va paralelo con el archivo de datos antes mencionado, pero realiza varias funciones mucho más importantes, como son: almacenar las claves públicas, verificar que se trata de una clave pública y no-privada, verificar que posea un correcto formato, cambiar la clave a un formato más compacto para minimizar el espacio usado en su almacenamiento, consultar por una clave, eliminar una clave, identificar el verdadero propietario de una clave y otras funciones que PGP posee, pero por no ser utilizadas por el servidor de claves no serán mencionadas. PGP utiliza un archivo especial llamado anillo de claves que lo emplea para almacenar las claves públicas (pubring.pgp) y se encuentra localizado en el mimo directorio que los archivos ".class" del servidor.

No está de más resaltar la importancia que tiene para la ejecución del servidor de claves el archivo de datos generales y el anillo de claves públicas ya que el servicio que ofrece se basa en la información que contienen estos archivos de allí que el servidor al iniciarse lo primero que hace es verificar si existe estos archivos y de no ser así no se inicia.

3.4 ANALISIS DEL SERVIDOR

Al iniciar el análisis del servidor se creyó prudente dividirlo en tres partes: descripción de actores, definición de objetos y el algoritmo de cada objeto. Esto fue necesario debido a que la implementación del servidor es orientada a objetos.

La descripción de actores juega un rol muy importante a la hora de la definición del entorno en el que se desenvolverá el servidor, pues aquí se establece de manera clara y explícita el papel que desempeña cada uno de las piezas que formarán el sistema Cliente-Servidor de claves públicas.

La definición de objetos y su algoritmo, a diferencia de la descripción de actores, tiene que ver con el funcionamiento interno del servidor de claves. Aquí se define que rol desempeña cada objeto dentro del servidor, el funcionamiento interno de cada objeto y su interacción con el resto de objetos todo esto de vital importancia a la hora de realizar una implementación orientada a objetos.

3.4.1 ACTORES

Cliente Administrador.

Descripción

1. Aplicación que interactúa con el servidor para la administración de claves públicas en el sistema

2. Cuando establece una sesión con el servidor, debe especificar su identidad

3. Envía al servidor requerimientos de adición, consulta, modificación y eliminación de claves públicas.

4. En caso de que una opción requiera información adicional acerca de datos del propietario de la clave, este cliente la proporciona también para que el servidor la procese.

Notas

•Para que el servidor responda a cualquiera de las opciones enviadas por el cliente administrador, este debió haber enviado previamente un user y un password registrados en el sistema.

•Sólo puede haber un cliente administrador conectado a la vez.

Cliente de correo

Descripción

1. Aplicación que interactúa con el servidor para la consulta de claves públicas, envío y lectura de correo electrónico.

Notas:

*Varios clientes de correo pueden iniciar una conexión con el servidor al mismo tiempo.

Aplicación PGP

Descripción

1. El servidor utiliza facilidades que esta aplicación brinda tanto para recuperar o eliminar claves públicas como para lograr eficiencia en el almacenamiento de las mismas.

Notas

•El servidor utiliza esta aplicación para la administración de claves públicas por la eficiencia que consigue en la manipulación de claves. La clave pública es de gran tamaño y almacenarla completa en un archivo seria ineficiente.

3.4.2 OBJETOS

startup realiza las validaciones correspondientes en la sintaxis de la línea de comando que arranca el servidor. Además revisa que los archivos necesarios para el servidor existan y por último minimiza el

tamaño del archivo de datos generales eliminando los espacios en blanco innecesarios.

ServidorPGP abre el socket en modo pasivo y crea hilos independientes por cada cliente que arriba al servidor.

ManejadorConeccion es el hilo independiente que atiende cualquier requerimiento del cliente. Se crea un nuevo hilo por cada cliente que arriba

ManejadorBD es el que interactúa con los datos. Es decir administra el archivo de datos generales e interactúa con PGP, mediante la ejecución de comandos.

ProtocoloApp es una clase muy importante en la comunicación entre todas las aplicaciones. Aquí se encuentran declaradas todas las opciones y estados que se utilizan en el protocolo de aplicación. Además su constructor puede recibir una cadena de requerimiento o respuesta y fraccionarla en sus diferentes campos.

LogPGP da soporte a la comunicación del servidor con la aplicación PGP, al interpretar al servidor los mensajes que PGP devuelve después de ejecutar algún comando.

3.4.3 ALGORITMOS

A continuación se detalla el funcionamiento interno de cada objeto, mediante la utilización de algoritmos. Cada objeto tiene dos secciones importantes y debidamente separadas que son los atributos o características del objeto y los métodos que son los procedimientos o funciones que agrupan el código a ser ejecutado por el objeto.

3.4.3.1 Clase startup

Atributos.

Métodos.

```
private startup (String argvs[]){
  analiza la línea de comando para determinar si
    ejecutar el servidor en modo debug y saber en
    que puerto arrancar (por defecto usa el 6001)
```

```
    verificarArchivos();
      nuevo servidorPGP();}

  private void verificarArchivos(){
    si no existe archivo de claves públicas terminar
    si no existe archivo de datos generales terminar
    leer linea a linea el archivo de datos generales
      si la linea es diferente a espacios en blanco
      guardar en nuevo archivo
      seguir leyendo hasta el final del archivo
    renombrar el archivo de datos generales a bak
    renombrar archivo nuevo a datos generales)

  private void usage(){
    Imprimir "Usar: java ServidorPGP.startup [-debug]
      [elPuerto]"}

  public static void main ( String argvs[] ) {
    nuevo startup(argvs);}
```

3.4.3.2 Clase servidorPGP

Atributos

```
public static int port
```

```
public static boolean  DEBUG
public static boolean admiInLine
public static long admiTime
public static String admiHost
public   static   final   String   ubicRelativa   =
  "ServidorPGP" + separadorPath
private ServerSocket sockServidor
private Socket sockCliente
private Thread elServidor
```

Métodos

```
public servidorPGP (){
  this.start();}

public void start(){
  abrir un socket para el servidor
  crear un hilo de ejecución independiente para el
    servidor
  arrancar ejecución del hilo servidor}

public void run(){
  mientras el hilo servidor no sea null
    poner al servidor a escuchar requerimientos
    nuevo manejadorConeccion (socket del cliente)
```

```
    stop()}

  public void stop (){
    cerrar socket servidor)
```

3.4.3.3 Clase manejadorConeccion

Atributos

```
  private static final int DEFAULT_SMTP_SERVER_PORT
    = 25
  private static final String SMTP_OK = "250"
  private static final String SMTP_MULTILINE_HELP
    = "214"
  private static final String SMTP_CLOSING = "21"
  private static final String SMTP_INTERMEDIATE_OK
    = "354"
  private static final int DEFAULT_POP3_SERVER_PORT
    = 110
  private static final String POP3_INTERMEDIATE_OK
    = "+OK"
  private Socket socketServidor
  private DataInputStream serverIn
```

```
private DataOutputStream  serverOut
private Socket socketCliente
private DataInputStream clienteIn
private DataOutputStream   clienteOut
private protocoloApp  solicitud
private String nameCliente
private int portCliente
```

Métodos

```
public manejadorConeccion (Socket  sock){
  abrir comunicación  por el socket
  this.start()}

public void run () {
  timeout de seguridad por si el administrador se
    queda colgado
  leer solicitud
  si es añadir
    si no esAdmin()
      devolver hay un administrador en linea
    crear objeto manejadorBD, ejecutar ingresar
    devolver resultado de ingreso
  si es modificar
```

```
  si no esAdmin()
    devolver hay un administrador en linea
  crear objeto manejadorBD, ejecutar consultar
  crear objeto manejadorBD, ejecutar eliminar
  crear objeto manejadorBD, ejecutar ingresar
  si no  puede ejecutar ingresar con los datos
    nuevos ejecutar  ingresar con  los datos
    viejos
  devolver resultado de modificación
si es eliminar
  si no esAdmin()
    devolver hay un administrador en linea
  crear objeto manejadorBD, ejecutar eliminar
  devolver resultado de eliminación
si es consultar existe
  si no esAdmin()
    devolver hay un administrador en linea
  crear objeto manejadorBD, ejecutar consultar
  devolver resultado de consulta
si es consultar todo del administrador
  si no esAdmin()
    devolver hay un administrador en linea
  crear objeto manejadorBD, ejecutar consultar
```

```
    devolver resultado de consulta
  si es consultar todo
    crear objeto manejadorBD, ejecutar consultar
    devolver resultado de consulta
  si es consultar clave
    crear objeto manejadorBD, ejecutar consultar
    devolver resultado de consulta
  si es verificar password
    Si adminInLine y no esAdmin()
      devolver hay un administrador en linea
    guardar DNS del cliente
    guardar el tiempo actual
    si no  filtro(user, password)
      devolver mal password
  si es cambiar password
    si no filtro(user, password)
      devolver mal password
    abrir archivo y guardar nuevo password
  si es puente SMTP
    puenteSMTP()
  si es puente POP3
    puentePOP()
  si es consultar con wildcards del administrador
```

```
    si no esAdmin()
      devolver hay un administrador en linea
    crear objeto manejadorBD, ejecutar consultar con
      wilds
    si no encuentra
      devolver no se encuentró
    por cada coincidencia  guardar en una cadena y
      al final devolver la cadena
  si es consultar  con wilds
    crear objeto manejadorBD, ejecutar consultar con
      wild
    si no encuentra
      devolver no se encuentró
    por cada coincidencia  guardar en una cadena y
      al final devolver la cadena
    si no es ninguno de los anteriores
      devolver solicitud desconocida
    cerrar el socket
    stop()}
public boolean filtro (String login, String  pass){
  si no coincide el login
    devolver falso
  leer archivo de password
```

```
    si no coincide el password
      devolver falso
    devolver verdadero}

  public void puenteSMTP(){
    abrir conexión con el verdadero servidor SMTP
    mientras el servidor no diga cerrar{
      leer del cliente comando
      mandar comando a el servidor SMTP
      leer del servidor SMTP resultado de comando
      mandar al cliente respuesta de comando}
  }

  public void puentePOP3(){
    abrir comunicación con POP3
    leer del servidor POP3 el mensaje
    escribir al cliente el mensaje
    hacer{
      leer comando del cliente
      mandar comando al servidorPOP3
      leer respuesta del servidorPOP3
      mandar respuesta de comando al cliente
    }mientras comano no sea QUIT
    cerrar sockets}
```

```
private boolean esAdmin(){
  si el ip de quien se logonea es igual al ip del
    cliente actual
    retornar true
  caso contario
    retornar falso}
```

3.4.3.4 Clase manejadorBD

Atributos

```
public static String ARCH_MAIN = "datos.dat"
```

Métodos

```
public Vector ingresar(String id, String clave,
  String datos){
  guardar la clave en un archivo cualquiera
  preguntar a PGP por el propietario de dicha clave
  si el identificador no es el email del usuario
    devolver clave mal
  si la clave tiene más de un identificador
    devolver identificador múltiple
  si la clave ya ha sido ingresada
    devolver clave duplicada
```

```
  ejecutar comando PGP para ingresar
  si pudo ingresar
    abrir archivo de datos generales y guardar los
      datos generales
    devolver clave ingresada
  caso contrario
    devolver no se pudo ingresar}

public Vector consultar(String id){
  ejecutar comando PGP para consultar
  si encontro la clave
    sacarla del archivo donde la puso PGP
    abrir archivo de datos generales  para completar
      la información
    devolver toda la información
  caso contrario
    devolver no se pudo consultar}

public int eliminar(String id){
  ejecutar comando PGP para eliminar
  si elimino la clave PGP
    eliminar los datos generales asociados
    devolver eliminación relizada
  caso contrario
```

```
    devolver no se pudo eliminar}

  public String ejecutar(String comando){
    ejecutar un comando como  un proceso
    recoger  la  respuesta  de  la  ejecución  y
      devolverla}

  public Vector consultarWild(String patron){
    abrir archivo de datos generales
    si el patron tiene @ buscar por e-mail
    caso contrario buscar por nombre
    leer archivo mientras haya lineas
      por cada coincidencia con el patron agregar a la
        cadena en la forma de nombre <e-mail>
    devolver la cadena con cero o mas datos}
```

3.4.3.5 Clase protocoloApp

Atributos

```
public static final int SOLICITUD = 0
public static final int RESPUESTA = 1
public static final String SEPARADOR = "::"
public static final int  OP_ANAD = 1
public static final int OP_MODI = 2
```

```
public static final int OP_ELIM = 3
public static final int OP_CONS_EXIS = 4
public static final int OP_CONS_TODO = 5
public static final int OP_CONS_TODO_ADMI = 6
public static final int OP_CONS_CLAV = 7
public static final int OP_VERI_PASS = 8
public static final int OP_CAMB_PASS = 9
public static final int OP_PUEN_SMTP =  10
public static final int  OP_PUEN_POP3 = 11
public static final int OP_CONS_WILD = 12
public static final int OP_CONS_WILD_ADMI = 13
public static final int ES_ERRO_DESC = -1
public static final int ES_ANAD_OK = 1
public static final int ES_MODI_OK = 2
public static final int ES_ELIM_OK = 3
public static final int ES_IDEN_EXIS = 4
public static final int ES_IDEN_NO_EXIS = 5
public static final int ES_IDEN_MAL = 6
public static final int ES_IDEN_MULT = 7
public static final int ES_CLAV_MAL = 8
public static final int ES_CLAV_DUPL = 9
public static final int ES_PASS_OK = 10
public static final int ES_PASS_MAL = 11
```

```
public static final int ES_CAMB_PASS_OK = 12
public static final int ES_ADMI_IN_LINE = 13
String email
String clave
String nombre
String telefono
int opcion
int estado
int tipo
```

Métodos

```
public protocoloApp(String cad, int tip){
  sacar de la cadena "cad" los diferentes campos}

public String siguieteToken() throws Exception{
  buscar un separador
  recuperar el campo hasta el separador
  reducir la cadena a un campo menos}

public String getEmail(){
  retornar email}

public int getOpcion(){
  retornar opcion}
```

```
public String getClave(){
  retornar clave}

public String getNombre(){
  retornar nombre}

public String getTelefono(){
  retornar telefono}

public int getEstado(){
  retornar estado}

public String getOtroEmail(){
  retornar clave}

public int getTipo(){
  retornar tipo}

public String getUser(){
  retornar email}

public String getPassw(){
  retornar clave}
```

```
public String getOldPassw(){
  retornar clave}

public String getNewPassw(){
  retornar nombre}

public String getSMTP(){
  retornar email}

public String getPOP3(){
  retornar email}

public String getLista(){
  retornar clave}
```

3.4.3.6 Clase LogPGP

Atributos

```
public static String IDEN_EXIS_1 = "Transport armor
  file:"
public static String IDEN_NO_EXIS = "Key not found
  in key ring"
public static String VIENE_USERID = "Key for user
  ID:"
```

```
public static int ESPACIO_ENTRE_TYPE_Y_USERID = 31
public static String IDEN_EXIS_2="1 matching key
  found"
public static String IDEN_EXIS_3="matching keys
  found"
public static String CLAV_MAL_1="ERROR: Bad ASCII
  armor checksum"
public static String CLAV_MAL_2="ERROR: Bad ASCII
  armor character"
public static String CLAV_MAL_3="No keys found"
public static String CLAV_DUPL ="No new keys or
  signatures in keyfile"
public static String CLAV_ANAD_OK ="1 new key(s)"
public static String CLAV_ELIM_OK = "Key removed
  from key ring"
```

Métodos

```
public static int ValIngresar(String Log,String
  Email){
  si CLAV_ANAD_OK existe en el string Log{
    si email igual al del Log
      retornar ingreso correcto
    caso contrario
```

```
    ingreso incorrecto
  }caso contrario{
    si existe CLAV_DUPL en Log
      retornar clave duplicada
    si existe CLAV_MAL_1 o CLAV_MAL_2 o CLAV_MAL_3
      en Log
    retornar clave mal
  }de no suceder nada de lo anterior retornar error
    desconocido}

public int ValConsultar(String Log,String Email){
  si IDEN_EXIS existe en el string Log{
    si EmailKxOk(Log, Email)
      retornar identificador existe
    caso contrario
      retornar identificador no existe
  }si IDEN_NO_EXIS existe en el string Log
    retornar identificador no existe
  si IDEN_EXIS_2 existe en el string Log{
    si EmailOk(Log, Email)
      retornar identificador existe
    caso contrario{
      si Email(Log,Email) es igual IDEN_MULT
```

```
        retornar identificador multiple
      caso contrario
        retornar identificador no existe}
  }caso contrario
    si existe IDEN_EXIS_3 en Log
      retornar identificador no existe
  de no  ocurrir nada de lo anterior reotornar
    error desconocido}

public int ValELiminar(String Log){
  si existe CLAV_ELIM_OK
    retornar eliminacion bien
  retornar error desconocido}

private boolean EmailKxOk(String Log,String Email){
  si no existe VIENE_USERID
    retornar falso
  verifica que en Log haya solo un identificador y
    que este coincida con la variable Email
    retornando falso o verdadero si se cumple ambas
    cosas}

private int EmailOk(String Log,String Email){
```

```
    verifica que en Log haya solo un identificador y
      que este coincida con la variable Email
      retornando falso o verdadero si se cumple ambas
      cosas}

public String EmailReal() {
  retornar esteEmail}
```

CAPITULO 4

DISEÑO DEL CLIENTE

4.1 FUNCIONALIDAD DEL CLIENTE

El proyecto consta básicamente de dos clientes totalmente diferentes en cuanto a la función que desempeñan. El primero es el cliente administrador, el cual lleva este nombre debido al mantenimiento que realiza sobre las claves públicas y los datos del dueño de la clave en el servidor de claves.

El segundo es el cliente de correo electrónico, como su nombre lo indica se encarga del manejo del correo como función básica, sin embargo la idea central de este cliente es la de proporcionar a los usuarios una

herramienta que les brinde la seguridad de que sólo el destinatario final leerá los mensajes que envíen. Para este fin la aplicación permite al usuario cifrar el texto de sus mensajes, utilizando el algoritmo de encriptación de PGP, antes de enviarlos. Además puede descifrar mensajes, para lo cual se necesita: que el mensaje esté encriptado con PGP, la clave secreta del destinatario y su frase de paso.

Sin importar de qué cliente se trate, se utiliza el mismo formato de mensaje para comunicarse con el servidor de claves. Este formato consta de la dirección de correo electrónico, seguido de una opción y de campos opcionales.

El desarrollo de un sistema que comunica aplicaciones a través de una red de computadoras usando arquitectura TCP/IP implica un requerimiento extra a la programación, en general, de cada una de las aplicaciones, que se refiere a la ubicación de los programas en la red. Las aplicaciones deben conocer la dirección IP y el puerto en el cual corre cada uno de los programas con los que deseen comunicarse.

Puesto que un applet solamente puede comunicarse con servidores que corran en la misma máquina desde la cual fueron descargados, el servidor y el código de las aplicaciones clientes deben estar siempre en la misma computadora, con lo cual, con sólo invocar una función del lenguaje JAVA™ que retorna la dirección IP de la misma, los clientes pueden conocer fácilmente la dirección IP del servidor.

Así mismo los clientes deben conocer el puerto en el cual está corriendo el servidor de claves PGP. Para dicho efecto se estableció como puerto default, el puerto 6001. Sin embargo, como se ha especificado anteriormente el servidor puede correr en cualquier puerto libre, esto es una decisión del administrador del servicio. Así pues este parámetro no puede ser una constante establecida en el código de los clientes sino más bien un argumento fácil de editar. Para dicho efecto el puerto se lo obtiene del tag con el que se invoca el applet del cliente en la respectiva hoja HTML.

4.1.1 FUNCIONALIDAD DEL CLIENTE ADMINISTRADOR

El cliente administrador es un applet (código JAVA™), es decir, está diseñado para operar en el entorno de un browser como Netscape o Internet Explorer.

El cliente administrador es de un solo hilo, no necesita más, ya que sólo existe un administrador.

Cada vez que el cliente administrador se conecta al servidor de claves: abre un nuevo socket, envia un requerimiento, recibe la respuesta y cierra el socket.

Antes de iniciar cualquier trabajo con las claves públicas, el usuario administrador debe ingresar su usuario y contraseña para el proceso de autenticación.

El cliente administrador recibe estos dos parámetros como una cadena de caracteres a la que se aplica una función que poseen todos los objetos en JAVA™ para hacer lo que se denomina "un hash", es decir, una especie de ciframiento único para cada combinación de caracteres. De esta manera cuando los envia por la

red hacia el servidor de claves, éstos viajan ilegibles.

De la misma manera viaja la nueva contraseña y su confirmación en caso de solicitarse un cambio.

Una que el usuario ha sido autenticado, aparece una pantalla en la que se debe ingresar el identificador de la clave pública ya sea que vaya a ingresarla por primera vez o que quiera consultarla para modificarla o eliminarla. Este identificador es la dirección de correo electrónico del propietario de la clave. Luego de ingresar el dato puede elegir entre ingresar, consultar o regresar.

Si decide ingresar una nueva clave pública, la dirección e-mail es enviada al servidor de claves para que éste certifique que no haya ninguna otra clave asociada a tal dirección. Si existe una, el cliente administrador es quien informa al usuario del problema mediante una ventana de diálogo, caso contrario aparece una nueva ventana donde el administrador debe ingresar los datos del propietario

tales como: nombre, número de teléfono y por supuesto la clave pública.

Una vez que se ingresó los datos, manda a grabar la nueva información enviando un nuevo requerimiento al servidor de claves junto con todos los datos del propietario de la clave. Luego espera la respuesta que puede indicar éxito o no, dependiendo de la validez de la clave.

En caso de que decida consultar, puede localizar cualquier clave pública en el servidor de claves, se puede hacer uso del "*" como comodín para realizar una búsqueda. Entonces la búsqueda puede hacerla con el identificador completo o en parte como por ejemplo *@*espol.edu.ec, que daría como resultado la lista de todos los que posean e-mail en la ESPOL. Para esto, el cliente administrador da el requerimiento al servidor de claves indicando si el identificador enviado es para una consulta completa directa o una búsqueda con comodines.

El resultado de la búsqueda, en caso de ser con comodines, aparece en otro panel con todos los ítems

(cada item consta del nombre y el e-mail) encontrados listados en orden alfabético ascendente del nombre del propietario de la clave. Asi, si quiere ver más detalles de determinado item debe hacer doble clic en el item o seleccionarlo y elegir "Detalles". En este momento se envia el e-mail, que se saca del item seleccionado de la lista, a la espera de los demás datos asociados a este identificador.

Una vez que los datos, nombre, número de teléfono y clave pública llegan se los muestra al usuario en otro panel donde puede modificarlos o eliminarlos definitivamente.

El resultado de la búsqueda, en caso de ser completa directa, aparece en el panel antes mencionado, donde el usuario puede modificar toda la información, a excepción del e-mail, o eliminarla (en este momento los datos no pueden editarse).

Si elige modificar, los datos pueden ser editados, pero se deshabilita la posibilidad de eliminarlos. Una vez realizado los cambios (inclusive puede cambiar la clave pública) puede elegir grabarlos. En

este momento el cliente administrador envía al servidor de claves un requerimiento que incluye todos los datos, y, se pone a la espera de recibir una respuesta.

Si elige eliminar, se envia un requerimiento que solo posee el identificador de la clave (el e-mail), y, se pone a la espera de la respuesta que puede ser de éxito o fracaso.

Cada vez que se va a cambiar la información de los archivos que el servidor de claves maneja, se pide una confirmación al usuario de que la acción a ejecutar está correcta.

4.1.2 FUNCIONALIDAD DEL CLIENTE DE CORREO

El cliente de correo electrónico es un applet (código JAVA™), es decir, está diseñado para operar en el entorno de browser como Netscape o Internet Explorer.

Este cliente utiliza un solo hilo para su conexión con el servidor de claves.

Hay que aclarar que el cliente de correo se comunica con el servidor de claves de diferentes maneras: Una, cuando abro un socket, envia el requerimiento de obtención de claves, recibe la clave y cierra el socket. Otra cuando le solicita al servidor de claves que opere como intermediario entre el servidor SMTP o el servidor POP3 y el cliente de correo, en este caso la conexión si mantiene la idea de sesión.

El cliente de correo elige a qué servidor SMTP o POPMAIL3 accesar.

Primero se tiene que configurar la aplicación de correo. Para ello la primera pantalla presenta toda la información requerida y necesaria para enviar o recibir correo.

Para poder enviar un correo nuevo, la aplicación necesita que se especifique el nombre y dirección de correo electrónico del remitente además del servidor SMTP, todo esto está en la parte llamada "Tu Identificación: ".

Para poder recibir correo, la aplicación necesita que se especifique el user y password de la persona que quiere ver sus mails además del servidor POP3 donde se supone tiene la cuenta, todo esto está en la parte bajo la frase "Necesario para recibir mails ".

En esta misma pantalla se puede elegir entre "Obtener mails" o "Enviar mail".

Al elegir "Enviar mails" se mostrará una nueva pantalla, en donde el usuario deberá llenar los siguientes campos:

To. Dirección (o varias direcciones separadas por ';') de correo electrónico de la persona (o personas) que recibirá(n) el mensaje.

Subject. Breve descripción del contenido del mensaje.

Mensaje. Texto a ser enviado.

Además dispondrá de opciones como:

Permitir envío de correo no encriptado. Este no es un botón sino un checkbox. Cuando está seleccionado (con

el visto) se puede enviar correo sin necesidad de que este esté encriptado. Si no está seleccionado, y el mensaje por alguna razón no ha sido encriptado aún, éste no puede ser enviado. Esto se debe a que la aplicación tiene como norma enviar sólo texto encriptado, a menos que el usuario especifique lo contrario explícitamente.

Consultar. Da paso a una nueva pantalla donde se puede realizar una búsqueda de las personas que constan en el servicio. La búsqueda puede hacerse con comodines de la misma forma como se explicó en la búsqueda del administrador, y los resultados aparecen en una lista de donde se puede escoger alguno con un doble click o en su defecto seleccionándolo y luego oprimir "llevar e-mail". En este momento el cliente de correo regresa al panel de envío, pero ahora con el texto del campo destinatario igual al e-mail recogido.

Enviar. Enviar el mensaje al destinatario o destinatarios. El envío se produce sin problemas si el texto está encriptado o si hay permiso para enviar

sin encriptar. En caso contrario se realizan los chequeos respectivos.

Cada vez que se envia correo electrónico, el cliente de correo solicita al servidor de claves un "puente" hacia el servidor SMTP.

Para el caso de envio de correo electrónico a varios destinatarios, se requiere que el texto se encripte de manera diferente para cada uno de los destinatarios, asi que lo que se envia definitivamente no puede ser una copia a todos, sino un correo diferente para cada uno. Por tal razón, se abre una sesión con el servidor SMTP por cada destinatario cuyo texto se encriptó.

Esto no sucede, si el texto se envia sin encriptar. Aquí si se trata de una copia para cada uno de los destinatarios, por lo que en la sesión con el servidor SMTP, se le indica que existen varios recipientes para el mismo mensaje.

La conexión con el servidor de claves termina una vez que la sesión SMTP termina.

Encriptar. Permite al usuario cifrar el texto utilizando la clave pública del destinatario proporcionada automáticamente por el servidor de claves públicas, en caso de que este la tenga, caso contrario se niega la encriptación.

Si hubiera más de un destinatario, se solicita la clave pública de cada uno de ellos. La conexión con el servidor de claves se mantiene hasta que no se reciba la respuesta respecto al requerimiento de clave pública para el último destinatario.

Si la aplicación no "encripta" algún mensaje para alguno de los destinatarios, éstos aparecen en una lista para información del usuario.

Cabe recalcar que el ciframiento del mensaje se hace en la máquina donde se está ejecutando el browser que permite correr el applet. Esto es porque todas las clases de "cryptix" encargadas de la encriptación son traídas por dicho browser. Con esto se asegura que el texto nunca salga de la máquina local sin haber sido encriptado antes, a no ser que el usuario indique lo contrario.

Limpiar. Limpia todos los campos de texto para ingresar nuevos datos.

Regresar. Regresa al panel previo.

Al elegir "Obtener mails" aparece otra pantalla siempre y cuando los datos necesarios para el correo entrante hayan sido ingresados en la configuración inicial.

Lo primero que se hace es solicitar un "puente" al servidor de claves para el servidor POP3, y, mantenerse en la sesión hasta obtener todos los mensajes por primera vez.

En esta pantalla aparecen listados todos los mensajes que traiga el cliente de correo. Cada ítem de la lista consta del nombre del remitente y del título del mensaje, y, por cada selección de estos ítems aparece el texto del mensaje en un área destinada para ello. El contenido del área cambia dinámicamente con la selección del ítem en la lista.

Para conservar los mensajes, todos han sido guardados temporalmente. Esto también ayuda a reconocer, si hay o no, nuevos mensajes en el servidor POP3.

Además mostrará los siguientes botones:

Desencriptar. Como ya se dijo, el contenido del área del mensaje cambia dinámicamente, y, asi mismo cambia la disponibilidad de este botón, es decir, solo está habilitado si en el contenido del mensaje existe texto encriptado con PGP.

Si se encuentra texto encriptado con PGP en los mensajes, para desencriptarlo se necesita tener la clave secreta del destinatario del mensaje y la frase de paso. Si todo sale bien puede ver el texto desencriptado en la misma área.

Obtener mails. Este botón hace lo mismo que el ubicado en la pantalla de configuración, pero con la diferencia que sôlo trae los mensajes nuevos.

Reenviar. Únicamente puede habilitarse este botón si no hay texto en el mensaje que esté encriptado, es

decir, para reenviarlo necesitaría desencriptarlo primero.

Se lo usa cuando alguno de los mensajes que estoy leyendo decido enviarlo a otra persona. Ahora bien, lo que hace esto es colocar el texto del mensaje en el área destinada para tal, en la pantalla de envio. El usuario a continuación puede ingresar el e-mail del nuevo destinatario e inclusive enviarlo encriptado si éste tuviese una clave pública en el servidor de claves.

Contestar. Unicamente se habilita este botón si no hay texto encriptado en el mensaje.

Se lo usa cuando quiero enviarle una respuesta a la persona que me escribió el mail que estoy leyendo. Igual que en reenviar, se pasa la pantalla de envio, pero esta vez el destinatario es obligatoriamente el remitente del mensaje que estoy respondiendo. El cliente de correo lo que hace es buscar el e-mail del mismo dentro del mensaje y colocarlo en el área de destinatario del panel de envio.

Regresar. Muestra el panel previo.

4.2 ANALISIS DEL SISTEMA CLIENTE ADMINISTRADOR

El diseño del Cliente Administrador, al igual que la implementación, es orientado a objetos. Por lo tanto, es necesario definir primero los actores que interactúan con el sistema como agentes externos al mismo. Luego debe definirse todos los objetos utilizados para la implementación con sus respectivos atributos y métodos. En las secciones siguientes se presentan el detalle de cada uno de estos tópicos.

4.2.1 Actores

Usuario Administrador

Descripción

1. Es el encargado de proporcionar la información necesaria para la ejecución de las opciones del cliente administrador.

2. Está encargado de garantizar la información que proporciona.

3. Se asegura que la comunicación cliente - servidor, en lo que tiene que ver al puerto, esté garantizada.

4. Es el único autorizado a ingresar información en el cliente administrador.

5. Ingresa los identificadores y patrones de búsqueda para el ingreso de nuevas claves y consulta de otras.

6. Realiza cambios en los archivos del servicio utilizando al cliente administrador, garantizando que los cambios sean correctos.

Servidor

Descripción

1. Aplicación que interactúa con el cliente administrador para la administración de claves públicas en el sistema.

2. Autentica el ingreso del usuario y contraseña que inicialmente se necesitan en el cliente administrador.

3. Verifica que solamente haya un cliente administrador conectado, enviando un estado de la conexión al cliente administrador.

4. Envía el estado de las respuestas a las solicitudes enviadas por el cliente administrador junto con los datos que fueron requeridos si ese es el caso.

4.2.2 Objetos

Administrador

El objeto Administrador es el applet donde se crea absolutamente toda la interface del usuario. Además es el que maneja la comunicación entre el servidor de claves y el cliente administrador.

La ejecución de sus diferentes métodos está supeditada a la acción o eventos que se produzcan en él.

4.2.3 Algoritmos

4.2.3.1 Clase Administrador

Atributos

```
Socket socket
DataOutputStream os
DataInputStream is
String host
String outBuffer
String  inBuffer
private boolean ConsultaPadre
private int indexItem
private int TAB
MainFrame top
protocoloApp protocolo
String EMail
int puerto
```

Métodos

```
void check_Clicked(Event event){
Si el estado de checkNewPassw es verdadero
  Habilitar campos de texto para ingresar la nueva
  contraseña y confirmarla.
```

```
Caso contrario
  Deshabilitar dichos campos.}
void aceptar_Clicked(Event event){
Limpiar displays.
Si los datos no están completos entonces regresar.
  Realizar un hash de user y password.
Si el estado checkNewPassw es verdadero
  Setear newpassword a la nueva contraseña
  Hacer un hash a newpassword
  Armar requerimiento con opción de cambio de
  contraseña.
Caso contrario
  Armar requerimiento con opción de verificación de
  contraseña.
  Llamar a conectar()
  Llamar a enviar(requerimiento)
  Llamar a recibir() y obtener respuesta
  Si el estado de checkNewPassw es verdadero
    Si la respuesta indica un cambio de contraseña
  OK
      Indicar cambio es OK
    Caso contrario
      Llamar a displayError()
```

```
  Caso contrario
    Si la respuesta es verificación de contraseña OK
      Mostrar Panell E-mail
    Caso contrario
      Llamar a displayError()
  Desconectarse}

void eliminar_Clicked(Event event){
   Mostrar pregunta "si o no" }

void eliminar_(Event event) {
 Llamar a conectar()
 Enviar requerimiento con opción de eliminar
 Recibir respuesta del servidor
 Si la respuesta es eliminar OK
  Si panel padre fue Consulta
    Eliminar item de la lista
 Caso contrario
  Llamar a displayError()
 Desconectarse  }

void grabarAdd_Clicked(Event event) {
Si no hay datos completos
  Regresar
```

```
Llamar a conectar()
Enviar requerimiento de añadir junto con la
clave,nombre y teléfono.
Recibir respuesta
Si respuesta es añadir OK
  Indicar adición OK
Caso contrario
   Mostrar displayError
Desconectarse}

void regresar_Clicked(Event event) {
Mostrar Panel0 Contraseña }

void modificar_Clicked(Event event) {
Deshabilitar eliminar
Mostrar  grabar modificar
Habilitar campos nombre, teléfono y clave}

void grabarMod_Clicked(Event event){
Mostrar pregunta "si o no"}

void grabarMod_(Event event){
Si no hay datos completos
  Regresar
```

```
Llamar conectar()
Enviar requerimiento de modificar junto con la
clave, nombre y teléfono.
Recibir respuesta

Si respuesta es modificar OK
 Indicar modificación OK
Caso contrario
 Mostrar displayError()
Desconectarse }

void regresarCon_Clicked(Event event){
Si se eligió modificar
  Mostrar pregunta si o no
Caso contrario
Llamar regresarCon_()}

void regresarCon_(Event event) {
 Si panel padre es panel1 E-mail
  Mostrar panel1 E-mail
Caso contrario
  Mostrar panel4 Buscar
Si lista de panel4 está vacia
    Deshabilitar detalles
```

```
Caso contrario
    Habilitar detalles y seleccionar primer item }

void detalles_Clicked(Event event){
 Extraer dirección Email de item seleccionado
 Llamar conectar()
 Enviar requerimiento para este Email con la opción
 de consultar todo dada por administrador
 Recibir respuesta
 Si respuesta es identificador existe{
  Obtener nombre, teléfono y clave para Email
  Mostrar panel3 Consulta con toda la información}
 Caso contrario
  Mostrar displayError()
 Desconectarse }

void regresarLis_Clicked(Event event) {
Mostrar panel1 E-mail }

void consultar_Clicked(Event event) {
 Si el identificador contiene asteriscos {
  Si contiene más de dos "*" {
    Mostrar displayError()
    Regresar }
```

```
  Si contiene al menos un asterisco
    opción = consulta con comodines }
Caso contrario
  Si identificador está vacio
    opción = consulta con comodines
  Caso contrario
    opción = consultar todos los datos
Llamar a conectar()
Enviar requerimiento (opción incluida) de consulta
al servidor.
Si se halló coincidencias
   Si se consultó con comodines {
     Mostrar panel4 Buscar
     Ordenar lista de identificadores obtenidos
     Presentar lista de identificadores hallados }
   Caso contrario {
      Mostrar panel de Consulta

      Presentar datos completos correspondientes a
       id dado }
Caso contrario
  Mostrar displayError()
Desconectarse... }
```

```
void regresarAdd_Clicked(Event event) {
Mostrar panel1 E-mail }

void anadir_Clicked(Event event) {
Si Email no tiene formato correcto
 Regresar
Llamar conectar()
Enviar requerimiento de consultar existencia
Recibir respuesta
Si respuesta es identificador no existe
  Mostrar panel2 Adición
Caso contrario
  Mostrar displayError
Desconectarse }

public synchronized void start(){
 Inicializar sockets
 Tomar parámetro puerto de hoja html
 Verificar puerto }

public synchronized void stop() {
 Cerrar sockets }

private void conectar(){
```

```
 Llama rendezvous(puerto) }

private void rendezvous(int port) {
 Abrir sockets en el puerto del host local }

public void enviar(String str){
Escribir en el socket
Si se produce un error{
 Mostrar displayError()
 Desconectarse } }

public String recibir(){
 Lee del socket
 Retorna lo leido
 Si se produce un error {
  Mostrar displayError
  Desconectarse } }

private static void quicksort(String[] a, int lo0,
int hi0) {
 Ordenar el arreglo "a" usando el algoritmo de
 ordenación "quicksort" y recurrencia  }
```

4.3 ANALISIS DEL SISTEMA CLIENTE DE CORREO ELECTRONICO

El cliente de Correo Electrónico, al igual que el Cliente Administrador fue diseñado e implementado utilizando técnicas de análisis y diseño orientado a objetos, para lo cual se analizaron los actores que intervienen en el sistema y se definieron los objetos necesarios para implementarlos. La sección siguiente muestra tanto la descripción de cada uno de los actores como la implementación de cada uno de los objetos con sus atributos y los algoritmos de cada uno de sus métodos.

4.3.1 Actores.

Servidor de claves

Descripción

1. Aplicación que recibe diferentes requerimientos del cliente de correo electrónico. Entre estos está el requerimiento de una conexión hacia el servidor POP3 o servidor SMTP especificado en la configuración. Una vez abierta la conexión, el

servidor de claves actúa como un "puente" transparente entre ambas aplicaciones.

2. El servidor de claves responde también a requerimientos de consulta de claves que hace el cliente SMTP para encriptar correo electrónico o la opción de consulta de claves incluida en el cliente de correo.

Usuario de Correo

Descripción

1. Proporciona la información necesaria para la ejecución de las opciones de correo electrónico.

2. Para obtener o enviar mails proporciona la configuración requerida para tales servicios.

3. Para desencriptar un mensaje proporciona la clave secreta y la frase de paso necesarias.

4. Para consultar proporciona el patrón de consulta que se enviará al servidor de claves

4.3.2 Objetos

Correo

El objeto correo es el applet donde se crea la interface con el usuario que será utilizada en el cliente de correo electrónico. Maneja todos los eventos que ocurren en todas las pantallas que aparecen a lo largo de las transacciones de correo.

Desencriptador

Este objeto provee un método que permite desencriptar un mensaje y mostrarlo con formato de texto. Para ello requiere la clave secreta, la frase de paso y el mensaje encriptado.

Para desencriptar este objeto emplea clases de cryptix disponibles en Internet.

Clave Dialog

Este objeto muestra una interface para que el usuario ingrese su clave secreta y su frase de paso. Luego

verifica si el formato de la clave ingresada coincide con el de una clave secreta pgp y constata que la clave secreta dada corresponda con la frase de paso proporcionada por el usuario. Luego de que la clave dada ha pasado esta verificación, este objeto utiliza el objeto desencriptador para descifrar un mensaje con la clave en cuestión.

Para las verificaciones de formato PGP y correspondencia de clave y frase de paso este objeto emplea las clases de cryptix.

4.3.3 Algoritmos

4.3.3.1 Clase Correo

Atributos

Socket sockConeccion

DataInputStream in

DataOutputStream out

private int PORT_PGP_SERVER;

String EMessage

DMessage

private boolean EMensaje

```
private Vector Correo,IDs
String NOMBRE, EMAIL, FROM
String  ServidorSMTP
Qsmtp connect
String  EMail, Para, opcion , outBuffer,
        inBuffer
boolean PopParent
int     numeroMensajes1, numeroMensajes2;
String  host, user, user2, password, ServidorPOP,
        server2=null;
boolean CambioUsuario,  todelete, borrados
public  Vector  Mensajes;
public static final int POP3PORT = 110;
private Socket socket
private DataInputStream is
private PrintStream os
private DataOutputStream out1
private String  Subject, De
private StringBuffer MsgText
```

Métodos

```
void GetMails_Clicked(Event event) {
    Si datos están completos
```

```
Si usuario y servidorPOP3 ha cambiado
  Cambio usuario = verdadero
Caso contrario
  Cambio usuario = falso
Mostrar panel para recibir mails
Llamar a startTransaction()
Si no hay mensajes
  Mostrar mensaje apropiado y retornar
Si Cambio usuario = verdadera //al inicio
                              Cambio usuario
                              es verdadera
  Crear arreglo Mensajes de dimensión (número
  de mensajes)
  Traer mails
  Añadirlos a vector y mostrarlos en la lista
Si Cambio usuario = falso
  Si hay nuevos mail
    Traer mails
    Añadirlos a vector
  Caso contrario
    Mostrar mensaje apropiado y retornar
Caso contrario
```

```
        Mostrar mensaje de datos incompletos y
        retornar }

void LimpiarMails(int numMesgs) {
        Vaciar lista de mails
        Crear nuevo vector de mensajes   }

void seleccionar(int nmensajes){
       Numero actual de mensajes = numero anterior;
        Llamar a popQuit();
        Si número de mensajes>0
          Seleccionar y mostrar mensaje 1
          Si validar_mensaje(texto de mensaje)
              Habilitar botón Desencriptar
              Deshabilitar botón Forward
              Deshabilitar botón Reply  }

void traer(int num1,int num2) {
  Para cada mensaje desde num1 hasta num2
    Traer mensaje
    Añadir subject y remitente a la lista
    Añadir contenido a Vector de Mensajes }

void  Desencriptar_Clicked(Event event){
```

```
  Si no se ha creado un objeto ClaveDialog:
      Crear un objeto ClaveDialog
  Objeto ClaveDialog.mostrar() }

public boolean validar_mensaje(String mensaje){
  Si mensaje contiene la cadena "----BEGIN PGP" y
  en una posición posterior la cadena "----END PGP"
    Retornar verdadero
  Caso contrario
    Retornar falso }

public String getMail(número de mail a obtener){
  Enviar comando "RETR " + número de mail a obtener
  Llamar a  getResponse(Stream de entrada)
  Mientras línea sea diferente de "."
    Leer mensaje por líneas e irlo formando, y
    obtener variables "Subject" y "De" de líneas que
    empiecen con "Subject" y "From", respectivamente
  Si ha sido requerido(variable borrar verdadero)
    Enviar comando  "DELE " + numero de mail
    Llamar a getResponse(Stream de entrada)
  Retornar mensaje leido}

public void popQuit(){
```

```
  Enviar comando "QUIT" a Sevidor Pop }

public String getResponse(Stream de entrada in){
  Leer linea de respuesta y separar primera palabra
  de palabras siguientes.
  Si linea comienza con "+OK"
    Retornar mensaje(palabras siguientes a +OK)
  Caso contrario
    Retornar null }

public int startTransaction() {
  Establecer conexión con el servidor de claves.
    Enviar requerimiento de puente POP3
  Si conexión es exitosa
    Enviar comando "USER" + (usuario)
    Enviar comando "PASS" + (password)
    Si (llamada a getResponse) devuelve null
      Mostrar mensaje de error de autenticación.
      Terminar
    Caso contrario
      Enviar comando "STAT" para obtener número de
      mensajes
      Retornar número de mensajes }
```

```
void NewMail_Clicked(Event event){
  Si nombre, e-mail y Servidor SMTP están completos
    Si e-mail contiene una '@'
      Mostrar Pantalla para enviar mails
    Caso contrario
      Mostrar mensaje de error
  Caso contrario
    Mostrar mensaje de error }

private void enviar(Event evt) {
  Si nombre, e-mail y Servidor SMTP están completos
    Establecer conexión con servidor de claves.
    Si no se estableció conexión
      Mostrar mensaje de error
      Salir
    Enviar requerimiento de puente SMTP
    Si hay varios destinatarios y mensaje no ha
    sido encriptado
      Si opción "Enviar mail sin encriptar" está
      marcada
        Enviar copia de mensaje a cada destinatario.
      Caso contrario
```

```
        Mostrar mensaje de error indicando los
        destinatarios a quien no se envió mail.
    Si hay varios destinatarios y se ha encriptado
    el mensaje
      Enviar mensaje a destinatarios para los cuales
      el mensaje ha podido ser encriptado.
    Si existen destinatarios para los cuales no se
    ha podido encriptar el mensaje
      Si opción "Enviar mail sin encriptar" está
      marcada
        Enviar mail sin encriptar a quienes no se
        haya enviado aún el mensaje.
      Caso contrario
        Mostrar mensaje de error indicando los
        destinatarios a quien no se envió mail
    Cerrar conexión. }

void consultar_Clicked(Event event)  {
  Si el identificador contiene asteriscos
    Si contiene más de dos "*"
      Mensaje de error de patrón de búsqueda
      Salir
    Si contiene al menos un asterisco
```

```
    opción = consulta con comodines
Caso contrario
  Si identificador está vacio
    opción = consulta con comodines
  Caso contrario
    opción = consultar todos los datos
Enviar requerimiento (opción incluida) de
consulta al servidor.
  Si se halló coincidencias
    Si se consultó con comodines
      Ordenar lista de identificadores obtenidos
      Mostrar lista de identificadores hallados
    Caso contrario
      Mostrar datos completos correspondientes a
      id dado
  Caso contrario
      Mensaje de error de "ninguna coincidencia
      con id dado" }

void Forward_Clicked(Event event) {
  Si configuración está completa
    Tomar el texto del mensaje leido y el Subject
    del mismo y colocarlos en los campos
```

```
    correspondientes de la interface para enviar
    mails. Subject precedido por "Fwd"
    Mostrar la interface para enviar mails.
  Caso contrario
     Mensaje de error de configuración }

void Reply_Clicked(Event event){
  Si configuración está completa
    Tomar el "From" del mensaje leido y el Subject
    del mismo y colocarlos en los campos
    correspondientes de la interface para enviar
    mails. Subject precedido por "Rply"
    Mostrar la interface para enviar mails.
  Caso contrario
    Mensaje de error de configuración }

void detalles_Clicked(Event event) {
  Tomar email de item seleccionado de la lista
  Enviar requerimiento de consulta con el email
  obtenido
  Leer respuesta del servidor
  Mostrar datos asociados al email seleccionado }
private static void quicksort(String[] a, int
lo0, int hi0) {
```

```
Ordenar arreglo "a" usando el algoritmo de
ordenación "quicksort", y empleando recurrencia.)

void consBut_Clicked(Event event) {
  Esconder pantalla para enviar mails
  Limpiar campos de pantalla consultar
  Mostrar Pantalla consultar }

void regresarDetCon_Clicked(Event event) {
  Esconder pantalla donde se muestran detalles de
  consulta
  Mostrar Pantalla principal de Consult; }

void ListMails_ListSelect(Event event) {
  Mostrar en el area de texto el mensaje
  correspondiente al item seleccionado en la lista
  Si validar_mensaje(texto)
    Habilitar botón desencriptar
    Deshabilitar botones Contestar y Reenviar
  Caso contrario
    Deshabilitar botón desencriptar
    Habilitar botones Contestar y Reenviar }

void regresarPOP_Clicked(Event event) {
```

```
  Esconder panel de recepción de mails
  Mostrar panel de configuración }

void regresarConsulta_Clicked(Event event){
  Ocultar panel de consulta
  Mostrar panel de envio de mails }

private String encode(String plaintxt, String
                      pubstr, String strseed){
  Con funciones de crytix obtener clave pública a
  partit de pubstr
  Obtener valor aleatorio en base a la hora
  Con este valor cifrar el mensaje
  Pasar texto cifrado a formato de texto
  Retornar mensaje cifrado }

private String obtenerClave (String id, String
                             host, int port){
  Formar un requerimiento para el servidor
  solicitando clave publica
  Llamar a función conectar.
  Enviar solicitud.
  Esperar repuesta del servidor
  Dividir respuesta en campos
```

```
  Si campo que trae estado de respuesta es igual
  a "ES_IDEN_EXIS"
    Retornar clave contenida en respuesta
  Caso contrario
    Retornar string "no clave"  }

void sndBut_Clicked(Event event){
  Si datos no están completos
    Retornar
  Si campo To  no contiene '@'
    Retornar
  Llamar a enviar(event); }

void sndEBut_Clicked(Event event){
  Si datos no están completos
    Retornar.
  Si campo To no contiene '@'
    Retornar.

  Mientras campo To contenga más direcciones email
  {
    Buscar clave pública de siguiente destinatario
    Si clave se encuentra
      Encode(texto de mail)
```

```
        Guardar texto en  arreglo de textos a enviar
        Cambiar botón "Encriptar" por "Desencriptar"
      Caso contrario
        Almacenar en arreglo texto original
    }
    Mostrar mensaje de error con destinatarios para
    los cuales no se pudo encriptar el mensaje   }

void sndDBut_Clicked(){
    Si datos no están completos
      Retornar
    Si opción "Permitir enviar mails sin encriptar"
    está habilitada
      Habilitar botón "Enviar"
    Caso contrario
      Deshabilitar botón "Enviar"
    Cambiar texto de mensaje por texto desencriptado
    Cambiar   botón   "Desencriptar"   por   botón
    "Encriptar" }

void clrBut_Clicked(Event event){
    Limpiar todos los datos de pantalla enviar mails
    Opción "Enviar mails sin encriptar" = false; }
```

```
void regresarSMTP_Clicked(Event event){
  Limpiar todos los campos de pantalla enviar
  Si pantalla anterior era la de Recibir mails
    Esconder pantalla Enviar mails
    Mostrar pantalla Recibir mails
  Caso contrario
    Esconder pantalla Enviar mails
    Mostrar pantalla Configuración    }

void checkEnviar_Clicked(Event event) {
  Si botón "Desencriptar" está mostrado
    Deshabilitar botón consulta
    Habilitar botón "Enviar"
  Caso contrario
    Habilitar botón consulta
    Si se marca opción "Enviar mail sin encriptar"
      Habilitar botón "Enviar"
    Caso contrario
      Deshabilitar botón "Enviar" }

public void enviarReqConsulta(String str){
  Escribir str en stream de salida
  Si ocurre error al escribir
    Mostrar mensaje de error apropiado    }
```

```
public synchronized void start(){
 Inicializar sockets
 Tomar parámetro puerto de hoja html
 Verificar puerto }

public synchronized void stop() {
 Cerrar sockets }

private void conectar(){
 Llama rendezvous(puerto) }

private void rendezvous(int port) {
 Abrir sockets en el puerto del host local }

public String recibir(){
 Lee del socket
 Retorna lo leido
 Si se produce un error
  Mostrar displayError
  Desconectarse }
```

4.3.3.2 Clase ClaveDialog

ATRIBUTOS

Correo parent

```
Label label,labelFrase,status
TextField laFrase
TextArea Clave
Button OKButton,CancelButton
desencriptador Desencriptador
```

METODOS

```
public void mostrar(){
  Mostrar el frame para ingreso de Clave y frase de
  paso)

public String getClave(){
    Retornar Clave)

public  boolean action(Event evt, Object arg){
  Si presiona botón Ok
    Si no ha ingresado clave o frase de paso
      Mostrar mensaje apropiado y retornar
    Caso contrario
      Ocultar frame de ingreso
      Crear  objeto  Desencriptador(Clave,  Frase  de
      paso)
```

```
        Llamar a métodos de desencriptador que
        verifican validez de clave
        Si clave inválida
          Mostrar mensaje de error en applet padre
          (parent) y retornar
        Caso contrario
          Desencriptar mensaje y añadirlo en área de
          texto de applet padre(parent)
          Deshabilitar botón de desencriptar en
          applet padre
        Si ocurre error al desencriptar
          Mostrar mensaje de error en applet
          padre(parent)
      Si presiona botón Cancel
        Ocultar frame de ingreso y retornar }
```

4.3.3.3 Clase desencriptador

ATTRIBUTOS

```
cryptix.crypt.rsa.SecretKey key
String eltexto
String lapassphrase
```

METODOS

```
desencriptador(String laclave,String lapassphrase){
  Verificar si laclave es una clave secreta pgp
  Si laclave no es una clave secreta pgp
    Lanzar excepción: "Clave no es una clave secreta
    PGP"
  Caso contrario
    Si laclave y lapassphrase no se corresponden
        Lanzar excepción "Frase de paso incorrecta"
    Caso contrario
      Almacenar clave en key  }

public final StringBuffer verPakete(
DataInputStream in ){
  Leer mensaje de in
  Verificar si mensaje fue encriptado con PGP
  Si mensaje no fue encriptado con PGP
    Lanzar excepción "Error descifrando..."
  Verificar si mensaje fue encriptado con clave dada
  Si mensaje no fue encriptado con clave dada
    Lanzar excepción "El mensaje no fue encriptado
    con la clave dada!"
  Caso contrario
  Descifrar mensaje y almacenarlo en descifrado
```

```
Retornar descifrado; }
```

4.3.3.4 Clase Qsmtp

ATRIBUTOS

```
static final int DEFAULT_PORT = 25;
static final String EOL = "\r\n";
protected DataInputStream reply
protected PrintStream send
protected Socket sock
```

METODOS

```
public Qsmtp( Socket hostid) throws
UnknownHostException, IOException {
    Llamar a Qsmtp(hostid, DEFAULT_PORT);    }

public Qsmtp( Socket hostid, int port) throws
UnknownHostException, IOException {
  sock = hostid;
  Obtener stream de entrada a partir de sock
  Almacenar stream de entrada en reply
  Obtener stream de salida a partir de sock
  Almacenar stream de salida en send
```

```
  Leer primera línea de comunicación usando reply
  Si línea no comienza con "220"
    Lanzar excepción de protocolo }

public Qsmtp( InetAddress address ) throws
IOException {
    Llamar a Qsmtp(address, DEFAULT_PORT);    }

public Qsmtp( InetAddress address, int port )
throws IOException {
   sock = Crear socket en address, port );
  Obtener stream de entrada a partir de sock
  Almacenar stream de entrada en reply
  Obtener stream de salida a partir de sock
  Almacenar stream de salida en send
  Leer primera línea de comunicación usando reply
  Si línea no comienza con "220"
    Lanzar excepción de protocolo }

public void sendmsg( String from_address, String
to_address, String subject, String message ){
  Si from_adrress tiene los caracteres "<" ">"
    Cortar de from_address lo que esté entre "<"
    ">" y almacenarlo en from
```

```
Caso contrario
  from = from_address
Verificar si la máquina en la que se está
corriendo tiene una dirección IP
  Si no se encuentra dirección Ip lanzar excepción
    "No se halló dirección IP"
Usando sock obtener nombre del host a contactar
Almacenarlo en host
Enviar comando "HELO" + host
Leer respuesta
Si respuesta no comienza con "250"
  Lanzar excepción de protocolo
Enviar comando "MAIL FROM" + from
Leer respuesta
Si respuesta no comienza con "250"
  Lanzar excepción de protocolo
Mientras to_addres tenga más direcciones
  Enviar comando "RCPT TO:" + from
  Leer respuesta
  Si respuesta no comienza con "250"
    Lanzar excepción de protocolo
Enviar comando "DATA" + from
  Leer respuesta
```

```
    Si respuesta no comienza con "354"
        Lanzar excepción de protocolo
  Enviar lineas
   "From:"  + from_address
   "To:"  + to_address
   "Subject"  + subject
   "Date:"+ msgDateFormat(fecha de hoy)
  Enviar lineas
    "Comment: Remitente sin autenticar"
    "X-Mailer: JNet Qsmtp"
    (linea en blanco)
    Mensaje
    "."
  Leer respuesta
  Si respuesta no comienza con "250"
        Lanzar excepción de protocolo }

public void close() {
  Enviar linea "QUIT" }

protected void finalize() throws Throwable {
  Llamar a close()}

private String msgDateFormat( Date senddate) {
```

```
Formar una fecha con dia, mes, año, hora, minutos,
segundos, diferencia con respecto a GMT
Retornar fecha formada  }
```

CAPITULO 5

MANUAL DEL USUARIO.

5.1 CLIENTE ADMINISTRADOR

Hay que recordar que un applet es cargado a través de una hoja html, es decir, una página capaz de correr en browser o un navegador de Internet. En el caso del Cliente Administrador dicha hoja llamada "admin.html" incluye un parámetro que es el puerto donde se comunica con el servidor de claves PGP. Por lo tanto, el usuario administrador puede modificarlo dependiendo del puerto que utilice el servidor de claves.

Aparte de esto se debe saber que no se necesita controlar la dirección IP del servidor de claves (necesaria para la comunicación, junto con el puerto), puesto que el código Java está en la misma máquina y por ende puede cónocer dicha dirección.

El Cliente Administrador es el encargado de ingresar nuevas claves públicas PGP, para que puedan ser accesadas por el servidor de claves, y de consultar, modificar y eliminar las existentes.

5.1.1 ACCESO PARA LA ADMINISTRACION

El cliente administrador antes de iniciar cualquier actividad en el programa debe "logonearse", es decir, debe ingresar su nombre de usuario y contraseña en el panel que se muestra en la figura 5.1, y luego, hacer click en el botón "Aceptar".

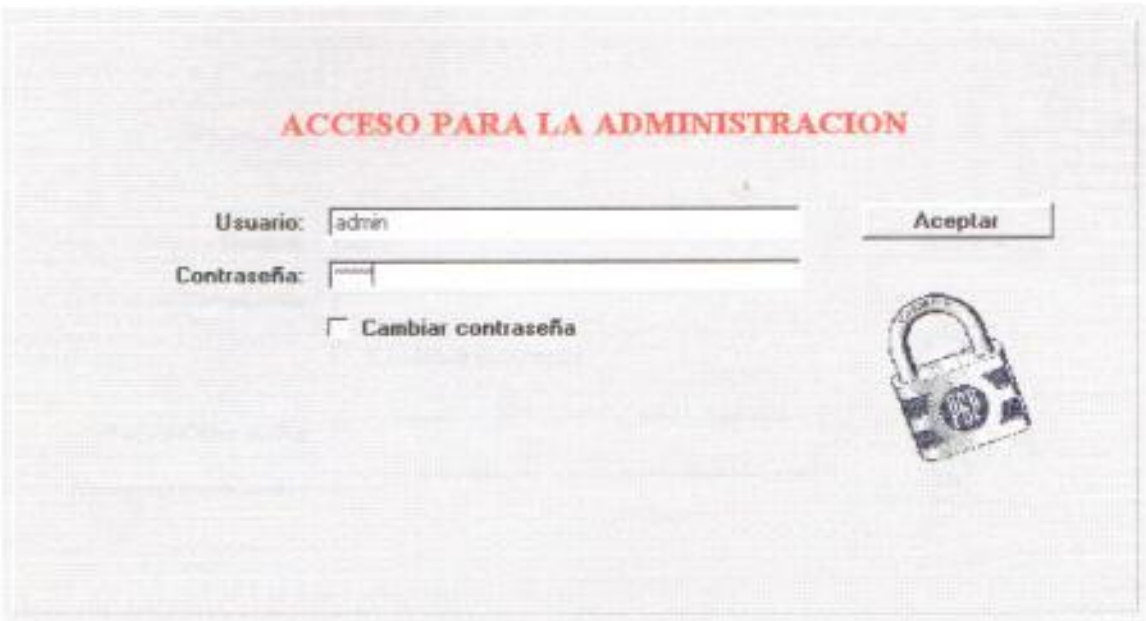


Figura 5.1.

La pantalla de acceso para la administración permite además, cambiar la clave del administrador señalando la opción "Cambiar contraseña".

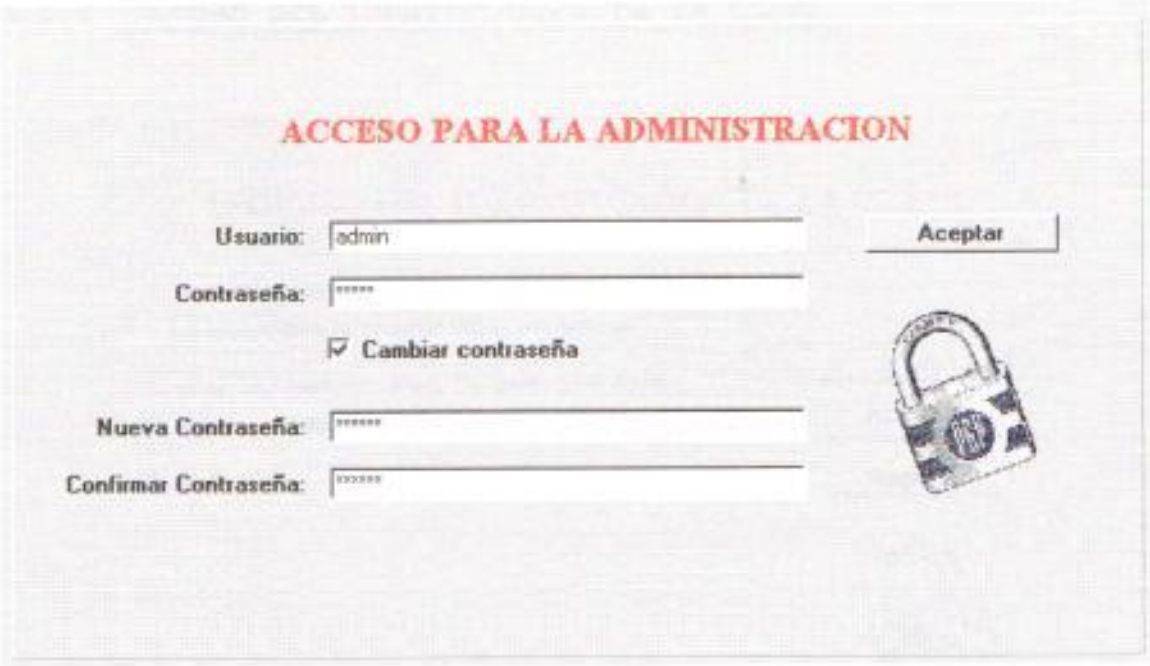


Figura 5.2.

Al elegir la opción "Cambiar contraseña" el administrador puede ingresar la nueva contraseña y confirmarla, tal como se muestra en la figura 5.2. Para realizar el cambio haga click sobre "Aceptar".

5.1.2 INGRESO DEL IDENTIFICADOR DE LA CLAVE.

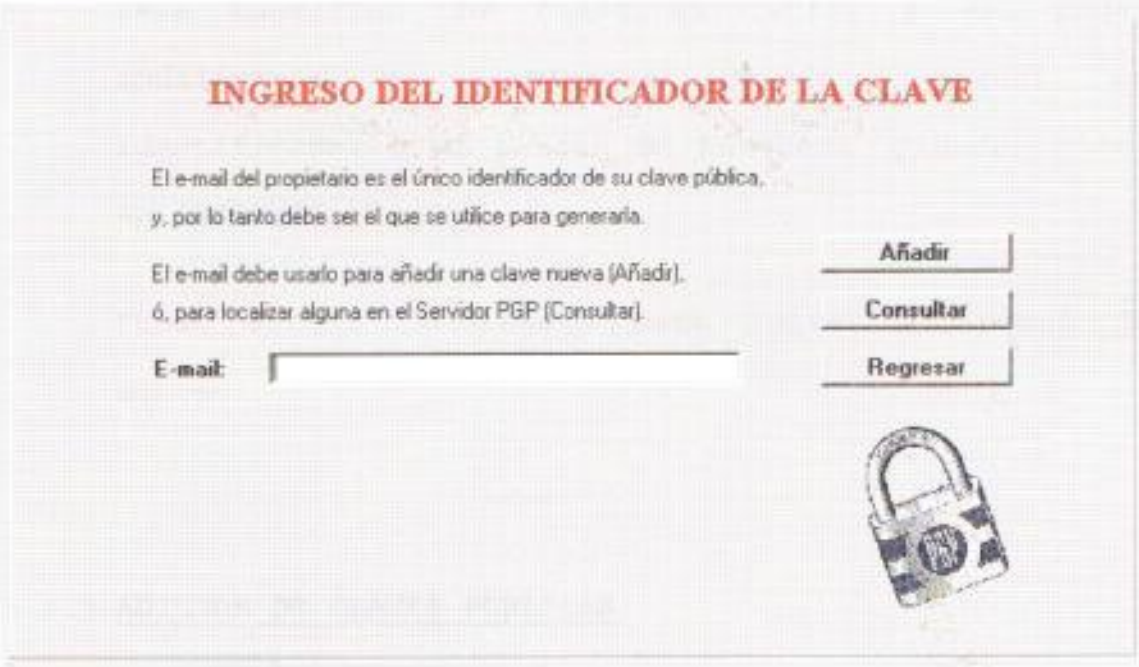


Figura 5.3.

Si la aplicación le dio acceso aparecerá un nuevo panel donde se pedirá ingresar una dirección de correo electrónico (ver figura 5.3).

Como ya se sabe, la dirección de correo electrónico es el único identificador de una clave pública.

Para añadir una clave nueva ingrese el identificador de la misma (dirección de correo electrónico del dueño) y haga click r".

en "Añadir".

Para consultar una clave existente, y así poder modificarla o eliminarla, puede ingresar un identificador o un patrón de búsqueda[1] y hacer click en "Consultar".

Puede elegir "Regresar" para volver al panel de acceso.

5.1.3 ADICION DE CLAVES PUBLICAS

Si se eligió la opción "Añadir" en el panel de la figura 5.3, y la dirección de correo electrónico tiene formato válido (incluye una "@"), aparecerá otro panel (ver figura 5.4) donde deberá ingresar el nombre completo del dueño de la clave, su teléfono y, por supuesto, su clave pública PGP.

---

[1] Véase patrón de búsqueda en la página 20

Si lo ingresado es correcto haga click en "Grabar" para almacenar la información.

Figura 5.4.

Todos los campos deben ser llenados, el programa no permite continuar si algún campo está vacío, lo mismo sucede si es que la clave que se pega no es una clave con formato PGP, es decir, que por lo menos empiece con "-----BEGIN PGP..." y termine con "-----END PGP...".

Puede elegir "Regresar" para volver al panel de ingreso del identificador.

5.1.4 CONSULTA DE CLAVES PUBLICAS

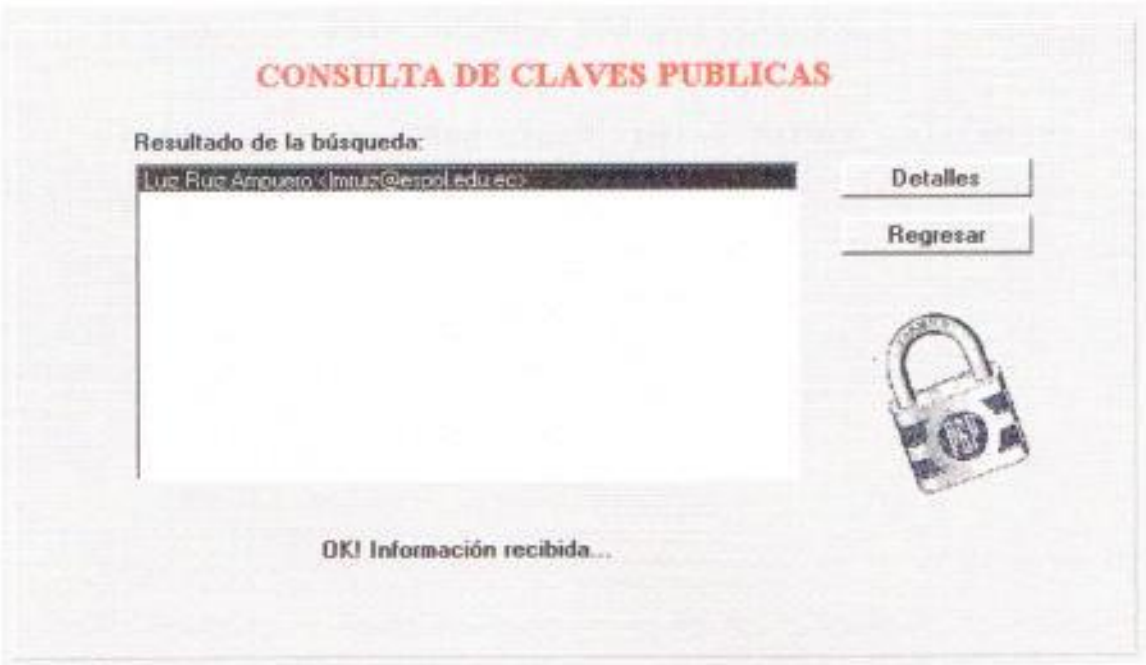


Figura 5.5.

Si se eligió la opción "Consultar" en el panel de la figura 5.4 con un patrón de búsqueda[1] aparecerá otro panel (ver figura 5.5) con una lista de todas las ocurrencias mostrando el nombre y dirección de correo electrónico del dueño de la clave. En este panel podrá elegir una ocurrencia y hacer click en "Detalles" para obtener más información.

Puede elegir "Regresar" para volver al panel de ingreso del identificador.

---

[1] Véase patrón de búsqueda en la página 20

5.1.5 MODIFICAR Y ELIMINAR CLAVES

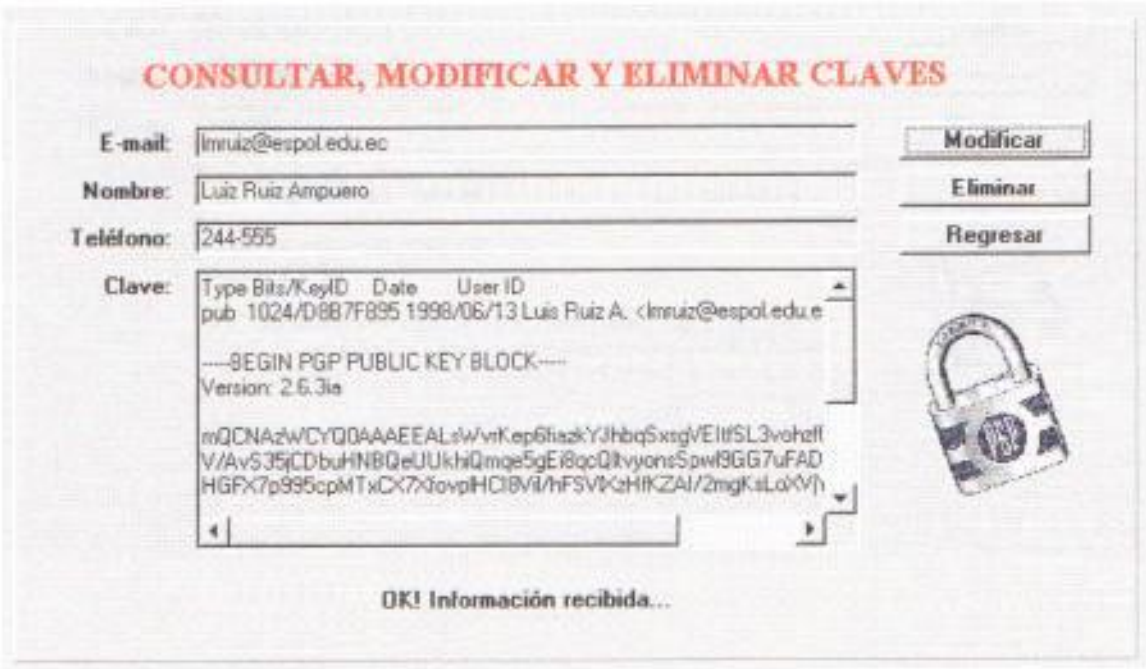


Figura 5.6.

Si se eligió la opción "Consultar" del panel de la figura 5.4 con una dirección de correo electrónico existente ó si se optó por "Detalles" del panel de la figura 5.5, aparecerá otro panel donde se podrá modificar o eliminar la información (ver figura 5.6).

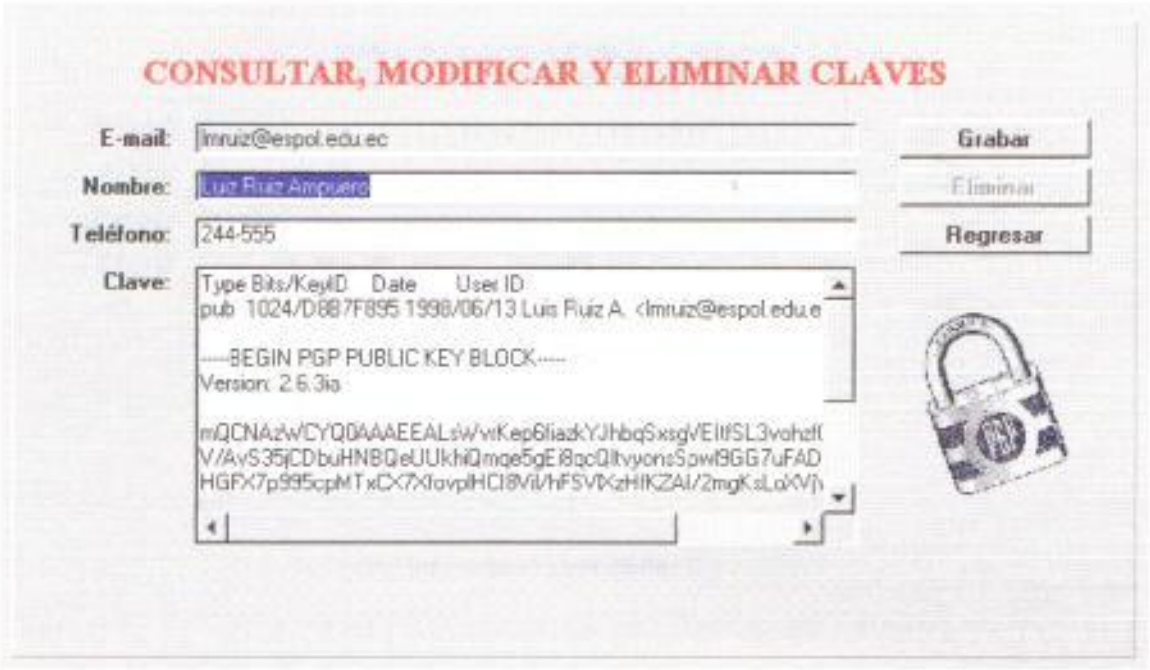


Figura 5.7.

Si se elige Modificar podrá cambiar la información del nombre, teléfono y clave, pero no la del identificador (dirección de correo electrónico).

Si desea guardar los cambios realizados debe hacer click sobre "Grabar" (ver figura 5.7).

Si elige Eliminar podrá borrar toda la información de la clave previa confirmación (ver la figura 5.8).

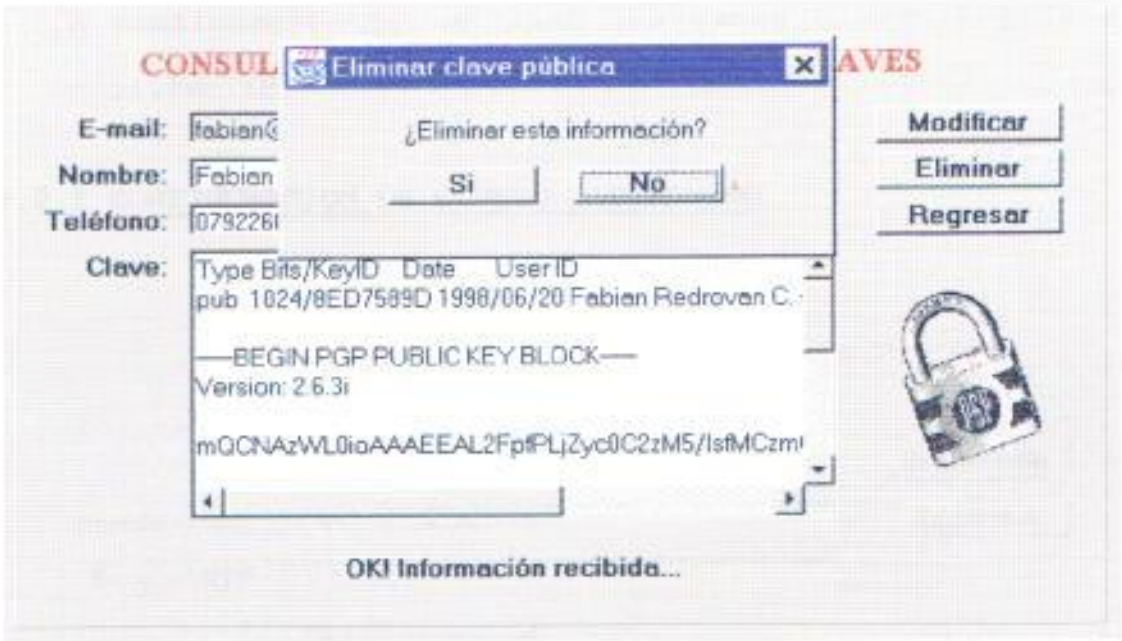


Figura 5.8.

Puede elegir regresar para volver al panel anterior ya sea el de ingreso del identificador (figura 5.3) o el de consulta (figura 5.5).

5.2 CLIENTE DE CORREO ELECTRONICO

Igual que en el Cliente Administrador, la hoja html del Cliente de Correo Electrónico tiene el mismo parámetro "puerto" modificable para comunicarse con el servidor de claves PGP.

Con esta aplicación se podrá enviar o recibir correo electrónico encriptado o no.

A continuación se detallan los pasos a seguir para su utilización.

5.2.1 CONFIGURACION DE CORREO ELCTRONICO

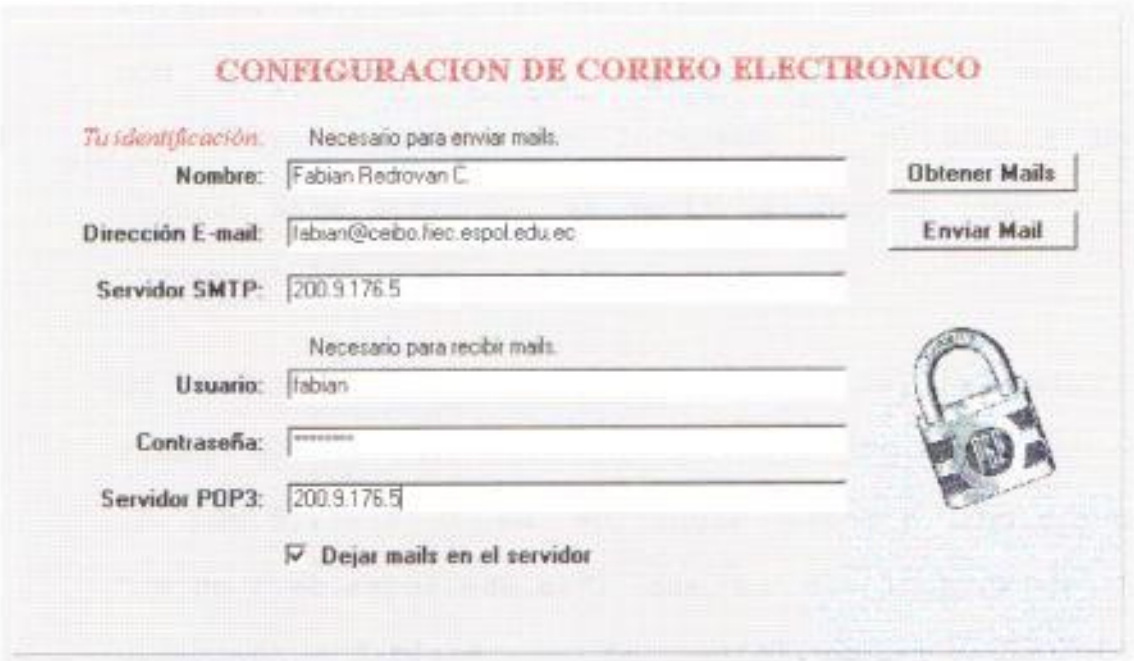


Figura 5.9.

Previa la utilización del cliente de correo electrónico es necesario configurarlo.

La configuración consta de dos partes, la información necesaria para enviar correo electrónico y la necesaria para recibir correo electrónico (ver figura 5.9).

Para poder enviar mails es necesario ingresar el nombre del usuario y su dirección de correo electrónico, conformando así la información del remitente. Además, se necesita especificar el servidor SMTP (Simple Mail Transfer Protocol) ya sea con su dirección IP (dirección de la máquina servidora de correo en Internet) o su nombre DNS (Domain Name Service), es decir, el nombre como se lo conoce en Internet.

En la figura 5.9 se puede apreciar estos tres campos, siendo el servidor SMTP especificado por su dirección IP 200.9.176.5 ó en su lugar haberse ingresado: "ceibo.fiec.espol.edu.ec", que es el dominio de la dirección "fabian@ceibo.fiec.espol.edu.ec" que fue especificada en el campo "Dirección E-mail".

En la información para recibir mails es necesario ingresar el nombre de usuario, la contraseña y el nombre del servidor POP3 (Post Office Protocol versión 3). Por lo tanto, es obvio que se debe tener una cuenta de correo electrónico en dicho servidor. El nombre de éste sigue las mismas reglas que el de SMTP.

Para que los mails permanezcan en el servidor luego de ser traidos, haga click sobre la opción "Dejar mails en el servidor".

Una vez que el cliente de correo electrónico esté correctamente configurado podrá obtener o enviar correo electrónico haciendo click sobre la opción deseada.

5.2.2 ENVIO DE CORREO ELECTRONICO

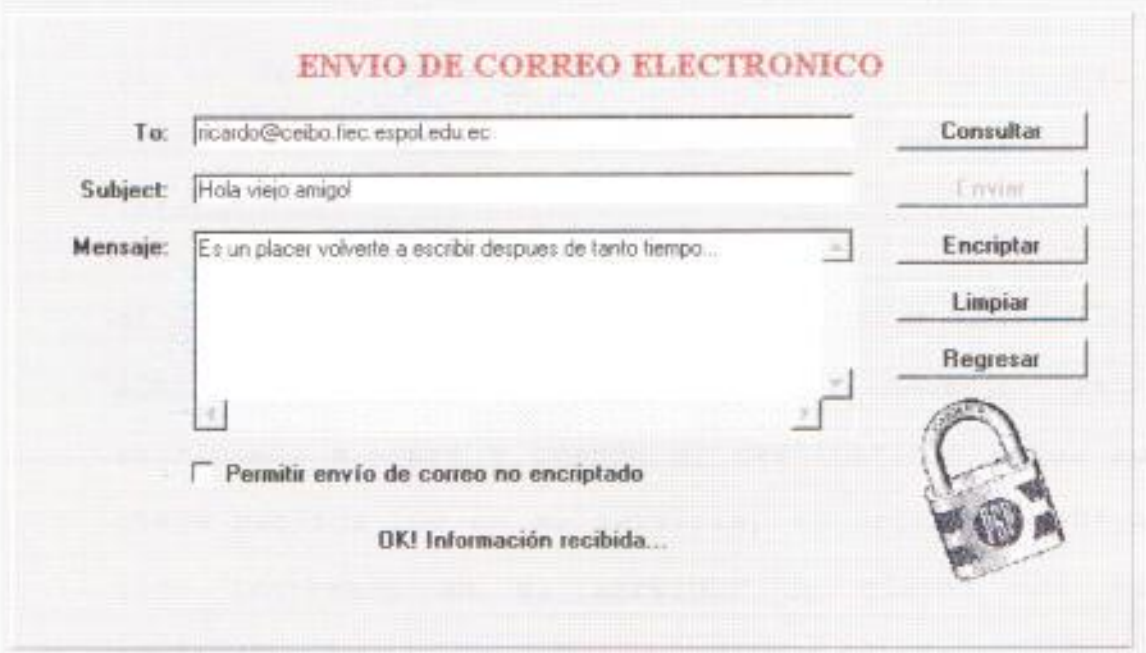


Figura 5.10.

Si se eligió "Enviar Mail" de la figura 5.9 aparecerá el panel que se muestra en la figura 5.10.

Para enviar un mensaje ingrese la dirección de correo electrónico destino, un tema y el mensaje que se desea enviar.

Para enviar el mensaje a varios destinatarios ingrese todas las direcciones destino separadas por ";".

Si se desea enviar el mensaje antes de encriptarlo, haga click sobre la opción "Permitir envio de correo no encriptado" y luego en el botón "Enviar".

Si se desea corregir todo lo que ha escrito haga click en "Limpiar" y vuelva a escribir la información.

Si lo que se desea es encriptar el texto haga click sobre el botón "Encriptar". El mensaje será encriptado siempre y cuando el destinatario tenga su clave pública PGP en el servicio, es decir, que haya sido ingresada en el servidor de claves por el administrador.

Si se logró encriptar el mensaje, éste se mostrará tal como en la figura 5.11.

Una vez encriptado el mensaje puede enviarlo haciendo click sobre "Enviar".

Si se desea corregir el texto del mensaje encriptado puede elegir "Desencriptar" y hacerlo.

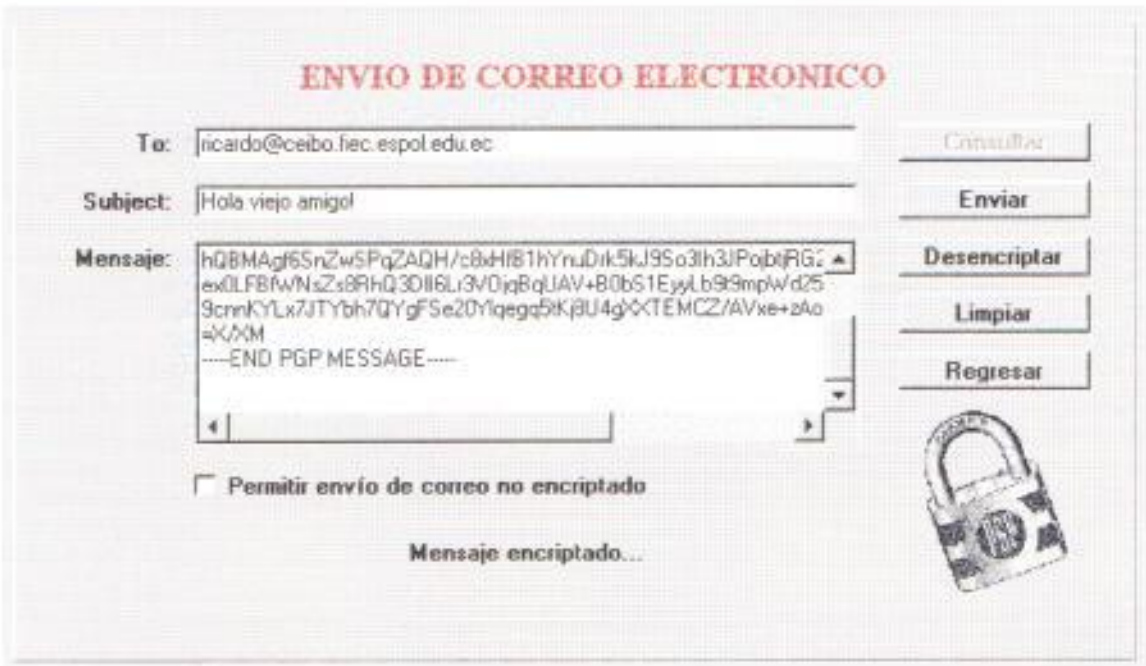


Figura 5.11.

El botón "Consultar" me permite buscar a quienes posean clave pública PGP en el servicio y así poder

enviarles correo encriptado. Al presionarlo aparecerá el panel mostrado en la figura 5.12.

En este panel se puede elegir "Buscar" para localizar ocurrencias de acuerdo al patrón de búsqueda[1] ingresado, tal como sucede en la consulta explicada anteriormente en el Cliente Administrador.

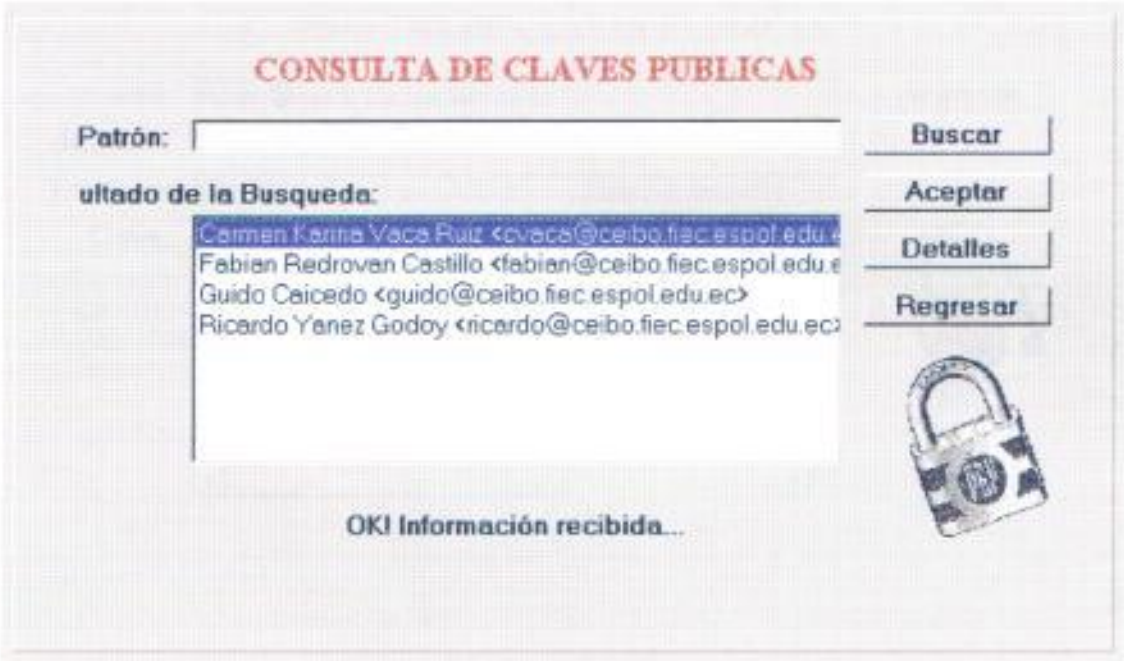


Figura 5.12.

De todas las ocurrencias, que constan de nombre y dirección de correo electrónica, se puede elegir una

---

[1] Véase patrón de búsqueda en la página 20

como destinatario y regresar al panel de la figura 5.10 haciendo click en "Aceptar", ó, ver más información haciendo click en "Detalles" (ver figura 5.13).

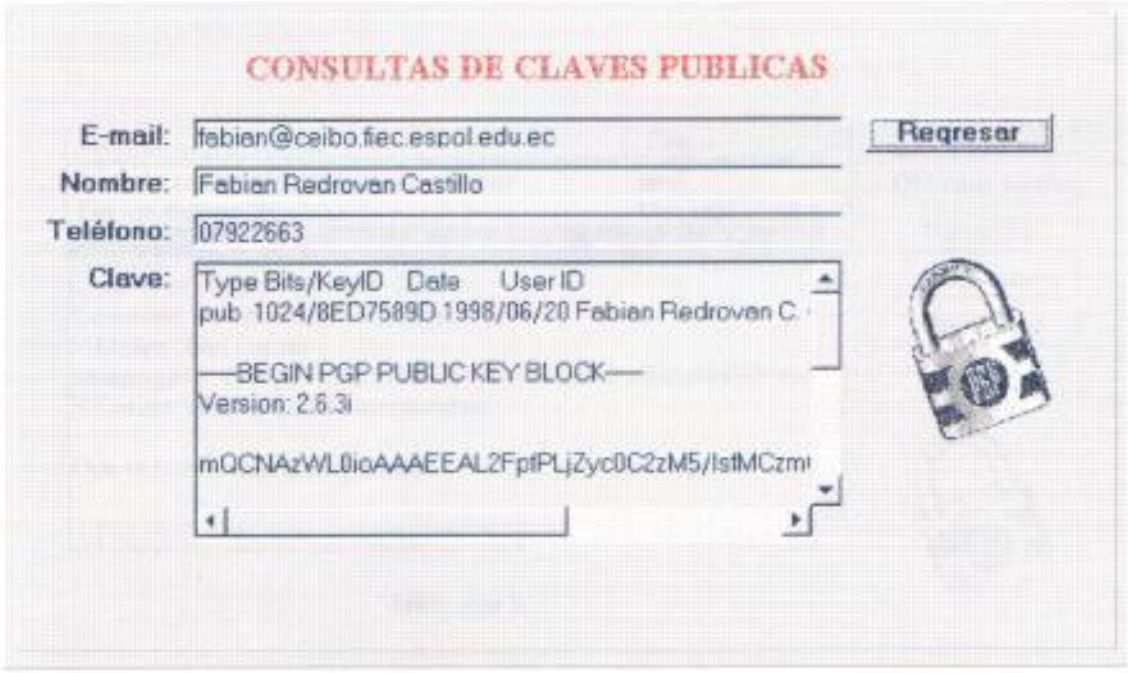


Figura 5.13.

Elegir "Regresar" en cualquier panel implica regresar al panel anterior.

5.2.3 RECEPCION DE CORREO ELECTRONICO

Si en el panel de la figura 5.9 se eligió "Obtener Mails" aparecerá el panel mostrado en la figura 5.14.

En esta pantalla se muestra la lista de los mails que le han llegado al usuario, para ver el texto del mensaje, selecciónelo y saldrá en la parte inferior ya sea que esté encriptado o no.

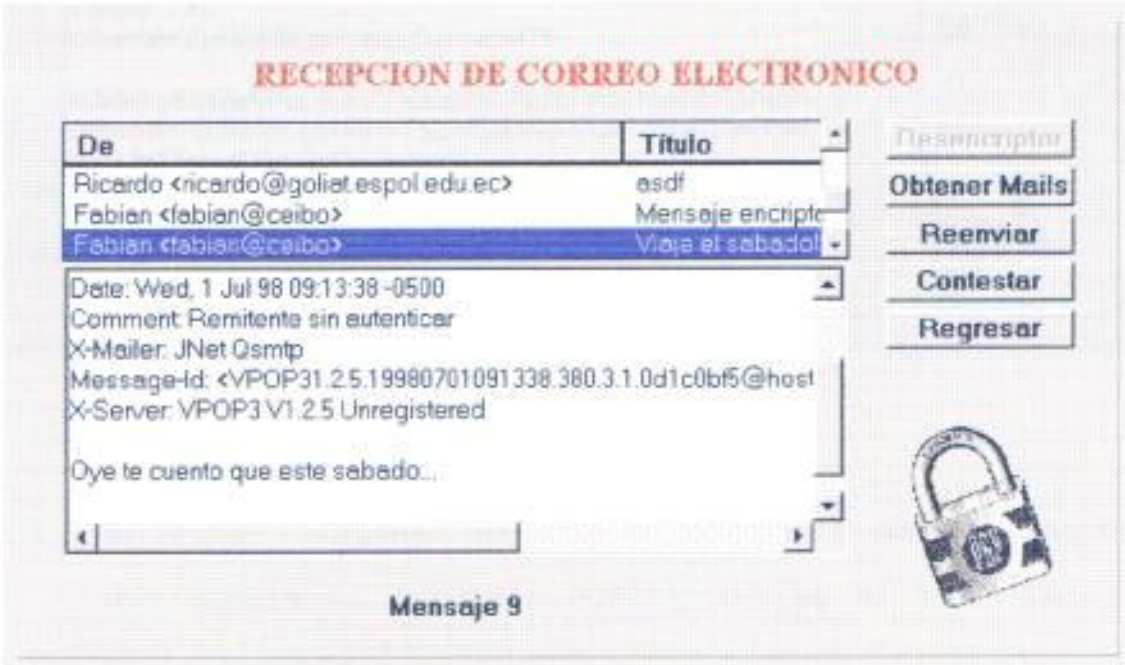


Figura 5.14.

Si el texto del mensaje está encriptado como en la figura 5.15 haga click en "Desencriptar". Esto hará aparecer una ventana de diálogo en la que se solicita la clave privada y la frase de paso para poder iniciar la desencriptación (ver figura 5.16).

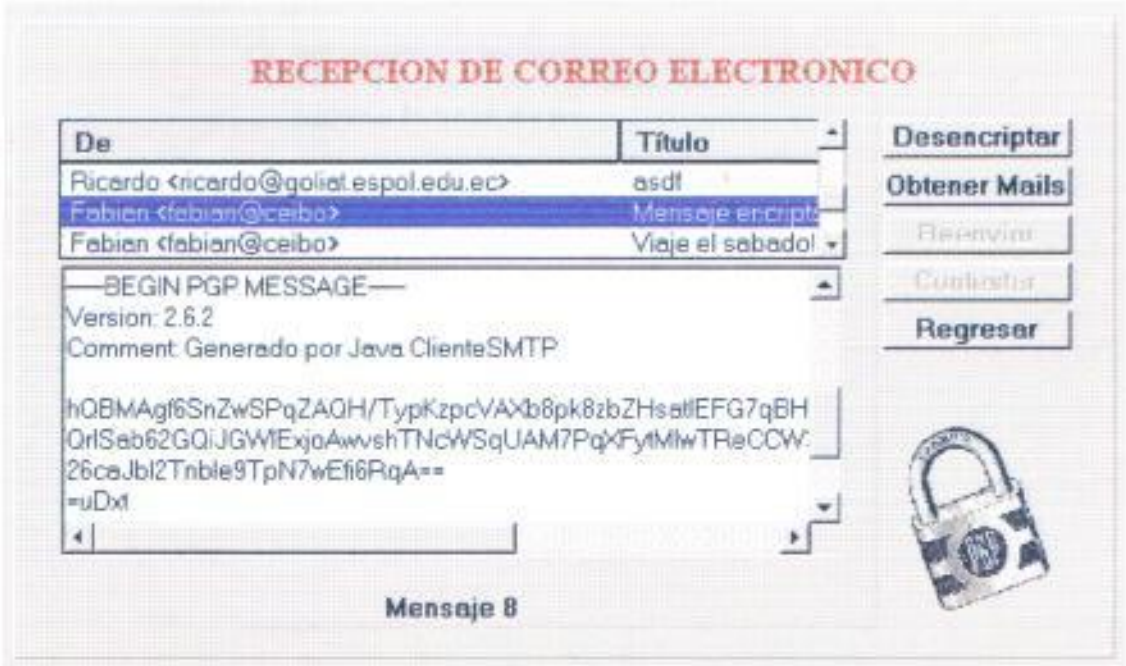


Figura 5.15.

El botón "Obtener Mails" le indica a la aplicación que accese al servidor POP3 y revise si han llegado mails nuevos.

Si el mensaje no está encriptado o si ha sido desencriptado, puede elegirse "Reenviar" (Forward) para que otra persona lo reciba, o puede elegir "Contestar" (Reply) para responder al remitente con un mensaje nuevo.

Figura 5.16.

Se puede elegir "Regresar" para ir al panel anterior, en este caso el de la figura 5.9.

5.3 MENSAJES DE ERROR

A continuación se detallan los mensajes de error en los clientes.

Para saber durante que pantalla (panel) suceden los mensajes de error, refiérase a las secciones 5.1 y 5.2, de acuerdo a los subtitulos.

5.3.1 CLIENTE ADMINISTRADOR

5.3.1.1 ACCESO PARA LA ADMINISTRACION

Los siguientes mensajes de error se producen durante la autenticación del usuario administrador en el panel de la figura 5.1.

Especificación de usuario.

Se ha presionando el botón "Aceptar" sin haber proporcionado el usuario.

Mensaje:
"Debe incluir el usuario para autenticación!"

Especificación de contraseña.

Se ha presionado botón "Aceptar" sin haber proporcionado la contraseña.

Mensaje:
"Debe incluir la contraseña para autenticación!"

Cambio de contraseña.

Se ha seleccionado "Cambiar contraseña", se ha presionado "Aceptar", y:

No se ha proporcionado nueva contraseña

Mensaje:

"Debe incluir la nueva contraseña!"

No se ha confirmado Nueva contraseña

Mensaje:

"Debe incluir la confirmación de la nueva contraseña!"

La nueva contraseña y su confirmación no son iguales

Mensaje:

"La nueva contraseña y su confirmación no coinciden!"

Autenticación.

El usuario y la contraseña proporcionada no corresponden al usuario registrado.

Mensaje:

"Login incorrecto."

5.3.1.2 INGRESO DEL IDENTIFICADOR DE LA CLAVE

Los siguientes mensajes de error se producen durante el ingreso del identificador de una clave en el panel de la figura 5.3.

Identificador de clave erróneo.

Se ha presionado "Añadir" proporcionando una dirección de correo electrónico que no contiene una '@'

Mensaje:
"Dirección e-mail no válida"

Se ha presionado "Añadir" proporcionando un identificador que ya está registrado en la base de datos del servidor.

Mensaje:
"Ya existe información para este identificador!"

Se ha presionado "Consultar" proporcionando un identificador que contiene más de dos '*'.

Mensaje:

"Patrón de búsqueda mal formado."

Se ha presionado el botón "Consultar" y no se han encontrado datos coincidentes con el patrón de identificador proporcionado.

Mensaje:

"No existe información para este identificador!"

5.3.1.3 ADICION DE CLAVES PUBLICAS

Los siguientes mensajes de error se producen durante la adición de una clave nueva en el panel de la figura 5.4.

Información a añadir incompleta.

Se ha presionado el botón "Grabar" y:

No se ha incluido el nombre del propietario de la clave

Mensaje:

"Debe incluir el nombre en la información!"

No se ha incluido el teléfono del propietario de la clave

Mensaje:

"Debe incluir el teléfono en la información!"

No se ha incluido la clave a ser añadida

Mensaje:

"Debe incluir la clave en la información!"

Problemas con la clave PGP.

Se ha presionado el botón "Grabar", el programa ha enviado los datos ingresados al servidor de claves y éste ha respondido que:

El identificador proporcionado por el administrador no corresponde al identificador utilizado al generar la clave que se está añadiendo.

Mensaje:

"Clave pertenece a " + {Identificador verdadero}

La clave que intenta añadir el administrador no tiene el formato de una clave generada por PGP.

Mensaje:

"La clave no tiene formato PGP."

La clave que se intenta añadir tiene otro propietario en el servidor de claves.

Mensaje:

"Clave duplicada de: " + (Identificador verdadero)

La clave proporcionada por el administrador tiene varios identificadores. Es posible que usando PGP se hayan añadido a la clave otros identificadores además de aquel con el que fue generada.

Mensaje:

"Esta clave tiene más de un identificador!"

5.3.1.4 MODIFICAR Y ELIMINAR CLAVES

Los errores al intentar grabar cambios en la información que se da en el panel de la figura 5.7, son del mismo tipo que en la adición (refiérase a la sección 5.3.1.3).

5.3.2 CLIENTE DE CORREO ELECTRONICO

5.3.2.1 CONFIGURACION DE CORREO ELECTRONICO

Los mensajes de error que a continuación se indican suceden en el panel de configuración (figura 5.9).

Especificación de usuario.

Se ha presionado el botón "Obtener Mails" en la pantalla de configuración sin haber proporcionado el usuario.

Mensaje:
"Usuario no especificado"

Especificación de contraseña.

Se ha presionado botón "Obtener Mails" sin haber proporcionado la contraseña.

Mensaje:
"Password no especificado"

Especificación de servidores.

Se ha presionado botón "Obtener Mails" sin haber proporcionado Servidor POP3

Mensaje:

"Servidor POP3 no especificado"

Se ha presionado el botón "Enviar Mails" sin haber especificado el servidor SMTP.

Mensaje:

"No ha especificado un servidor SMTP!"

Identificación incompleta.

Se ha presionado el botón "Enviar Mails" sin haber especificado datos de quien envía el correo:

El Nombre,

Mensaje:

"No ha especificado su nombre!"

La dirección electrónica,

Mensaje:

"No ha especificado su dirección e-mail!"

5.3.2.2 RECEPCION DE CORREO ELECTRONICO

Durante la obtención de los mensajes (figura 5.14) pueden surgir mensajes de error. A continuación algunos de ellos.

Autenticación.

El Servidor POP3 ha respondido "Login incorrecto". Debe regresar a la pantalla de configuración y modificar el campo user o el campo password.

Mensaje:
"Login incorrecto ... Revise configuración"

Interacción con servidor POP3.

El cliente ha enviado un comando al servidor POP3 y éste ha respondido "-ERR" que significa error al procesar el comando.

Mensaje:
"Error en el proceso de comandos"

Desciframiento.

El usuario ha proporcionado una clave que no corresponde a una clave secreta de PGP. Puede ser que se haya proporcionado una clave pùblica

Mensaje:
"La clave no es una clave secreta pgp!"

La clave no tiene un formato correcto.

Mensaje:
"No se pudo leer la clave!"

La frase de paso no corresponde a la clave secreta proporcionada.

Mensaje:
"Frase de paso incorrecta!"

El texto del correo electrónico no contiene un mensaje encriptado con PGP

Mensaje:
"El mensaje no fue encriptado con PGP!"

Hubo algún error mientras se desencriptaba el mensaje

Mensaje:

"Error descifrando..."

El texto del correo electrónico contiene un mensaje encriptado con una clave pública PGP que no corresponde a la clave secreta suministrada por el usuario

Mensaje:

"El mensaje no fue encriptado con la clave dada!"

5.3.2.3 ENVIO DE CORREO ELECTRONICO

Durante el envio de correo (figura 5.10) pueden surgir los siguientes mensajes de error.

Datos incompletos.

Se ha presionado el botón enviar y los datos no están completos:

Falta campo "To",

Mensaje:

"Debe incluir la dirección del destinatario!"

Falta campo "Subject",

Mensaje:

"Debe incluir un tema para el mensaje!"

Dirección destinatario no válida.

Se ha presionado el botón enviar habiendo colocado una dirección de correo inválida en el campo "To".

Mensaje:

"Dirección " +(dirección dada)+ " no es válida"

Mensaje no encriptado.

Uno o varios mensajes no han sido encriptados por cuanto la clave del destinatario(s) no se ha encontrado en el servidor de claves

Mensaje:

"Destinatarios cuyos mensajes no fueron encriptados" + (LISTA DE DESTINATARIOS)

Mensaje no enviado.

Uno o varios mensajes no han sido enviados por cuanto no fueron encriptados previamente y no se ha dado permiso de envio de correo sin encriptar.

Mensaje:

"Destinatarios cuyos mensajes no fueron enviados" + {LISTA DE DESTINATARIOS}

Servidores.

Al intentar enviar se ha producido un error por cuanto el servidor de claves no responde

Mensaje:

"No se puede contactar servidor PGP."

El servidor SMTP especificado en la configuración no responde al requerimiento hecho por el cliente

Mensaje:

"Servidor SMTP " +{ServidorSMTP}+ "no responde."

5.3.3 OTROS MENSAJES DE ERROR

Los siguientes mensajes de error pueden ocurrir en cualquiera de las instancias de ambos clientes.

Errores en la comunicación.

Se ha producido algún error en el manejo de funciones que controlan la comunicación de red entre cliente y servidor.

Se desea entablar comunicación con el host del servidor de claves PGP y éste no se encuentra:

Mensaje:
"No se puede encontrar el host: " + host

El servidor de claves no responde:

Mensajes:
"Asegúrese que el servidor esté levantado."

"No se recibieron datos del servidor."

Ocurrió un error en el intercambio de comandos con el servidor de claves.

Mensaje:
"Error desconocido!. Inténtelo nuevamente."

Hubo algún problema en la red, con el sistema operativo de la máquina u otro problema desconocido que no permita la comunicación:

Mensajes:

"No se puede utilizar el socket..."

"No se puede escribir en el socket."

"Stream de salida falló!"

"Servidor desconectado! Completando desconexión."

"No se puede leer del socket!"

"No se puede iniciar el socket."

CONCLUSIONES Y RECOMENDACIONES

Java es una herramienta de programación orientada a objetos muy poderosa debido a su gran flexibilidad, transportabilidad y facilidad, junto a la aplicación PGP, tan ampliamente difundida para diversas plataformas y de fácil manejo, constituyeron las mejores herramientas para el tipo de trabajo realizado.

Con la utilización del paradigma de par de claves asimétricas se ha logrado cumplir uno de los principales objetivos planteados: Suplir la deficiencia de seguridad en el transporte de correo a través de la Internet, recayendo dicha seguridad en la responsabilidad con que los propietarios manejen sus claves.

La comunicación entre servidores y clientes es posible gracias a la arquitectura de comunicación TCP/IP, que ha alcanzado una envidiable popularidad gracias a su eficiencia puesta a prueba día a día en la Internet.

El empleo de esta arquitectura ahorra gran cantidad de trabajo a quienes desarrollan aplicaciones para Internet, puesto que el programador se concentra únicamente en el desarrollo y optimización de la aplicación olvidando problemas como ruteo de paquetes a través de la red, timeout, etc., puestó que estos problemas son abordados por las capas interiores de la arquitectura y por la aplicación, que es donde el programador puede poner todo su esfuerzo.

Recomendamos ampliamente la utilización de lenguajes orientado a objetos, por la reusabilidad del código que brinda, la organización que se obtiene en el código resultante da facilidad a la hora de realizar cambios, por esto y otras gratas experiencias podemos decir que este tipo de lenguajes dominará el futuro de la programación.

APÉNDICES

APÉNDICE A

CRIPTOGRAFÍA

CÓMO FUNCIONA PGP

Para entender detalles importantes de este proyecto sería de ayuda estar familiarizado con el concepto de criptografía de llave pública (clave pública). En cualquier caso, he aquí unas cuantas observaciones como introducción.

En primer lugar, algo de vocabulario básico. Supongamos que quiero enviar un mensaje que nadie excepto el destinatario pueda leer. Podría "encriptar" o "cifrar" el mensaje, lo que significa revolverlo de una forma tremendamente complicada, con el fin de que resulte ilegible para cualquiera que no sea el destinatario original del mensaje. Se pone una "clave" criptográfica para encriptar el mensaje y el destinatario tiene que utilizar la misma clave para descifrarlo o "desencriptarlo". Por lo menos así funciona en los criptosistemas funcionales de clave única.

En los criptosistemas de llave pública, todo el mundo tiene dos llaves, una revelada públicamente y otra secreta

(llamada también llave privada). Cada llave abre el código que produce la otra. Saber la llave pública no sirve para deducir la llave secreta correspondiente. La llave pública puede publicarse y distribuirse pero no asi la llave privada que la debe tener el propietario en un lugar seguro. Este esquema de parejas de claves (claves asimétricas) proporciona intimidad y seguridad sin necesidad de utilizar algún canal seguro para distribuir la única clave que usan los criptosistemas convencionales.

Cualquiera puede utilizar la llave pública de un destinatario para encriptar un mensaje y el destinatario empleará su llave secreta correspondiente para desencriptarlo. Sólo él podrá hacerlo, porque nadie más tiene acceso a esa llave secreta. Ni siquiera la persona que lo encriptó podria descifrarlo.

Las llaves públicas se guardan en "anillos de clave" que incluyen el identificador de usuario del propietario (el nombre de esa persona [y algún dato único, como la dirección de correo electrónico]), un sello de hora del momento en el que se generó el par y el material propio de la clave. Cada llave secreta tiene su propia contraseña,

por si alguien roba la clave. Cada archivo ("anillo") de claves contiene uno o más de estas llaves.

Las llaves se identifican internamente mediante un "identificador de llave", que es una "abreviatura" de la llave pública (sus 64 bits menos significativos). Cuando se muestra este identificador, sólo aparecen los 32 bits inferiores para mayor brevedad. Aunque muchas llaves pueden compartir el mismo identificador de usuario, a efectos prácticos no hay dos llaves que compartan el mismo identificador de llave.

El destinatario utiliza el identificador de la llave para buscar la llave pública de la persona a la que se le enviará correo. El programa busca automáticamente la llave pública y el identificador de usuario en el archivo de llaves correspondiente (pubring.pgp).

Los archivos cifrados llevan como prefijo el identificador de la llave pública con la que se han encriptado. El destinatario busca este prefijo de identificación para encontrar la llave secreta y poder desencriptar el mensaje. Su programa busca automáticamente la llave secreta en el archivo de llaves correspondiente (secring.pgp).

Estos dos tipos de archivo constituyen el método principal para almacenar y gestionar las llaves públicas y secretas. En lugar de mantener las llaves individuales en archivos separados, se reúnen en llaveros para facilitar la búsqueda automática, ya sea por identificador de llave o por identificador de usuario. Cada usuario mantiene su propio par de llaveros. Las llaves públicas individuales se guardan en archivos aparte durante el tiempo necesario para enviarlas a algún amigo, que las añadirá entonces a su propio llavero. Por tal razón es imprescindible que estos dos archivos existan en la máquina servidora junto con el programa PGP.

En resumen, las claves públicas que se van a gestionar constarán en el archivo destinado para tal fin: "pubring.pgp". El servidor debe indicarle a PGP que agregue claves a tal archivo, que remueva del mismo de forma definitiva o que las extraiga simplemente para ser proporcionadas. Cabe recalcar que debido a lo explicado anteriormente no se puede cambiar la clave pública de un usuario sin alterar su identificador, por tal razón si se

intenta hacer esto, primero se debe eliminar la anterior y luego agregar la clave nueva.

Adición de una clave al llavero

Cuando se quiere añadir una clave al llavero (público o secreto), se necesita que ésta se encuentre en un archivo cualquiera. Este archivo puede también tener más de una clave y PGP se encargará no solo de separarlas e identificarlas sino también de verificar su validez y de que no estén duplicadas en el archivo. En cualquier caso PGP informa el resultado, que puede ser direccionado a un archivo.

La línea de comandos que utiliza PGP para tal fin es:

```
pgp -ka fdclaves [llavero]
```

El nombre del llavero por omisión es "pubring.pgp" o "secring.pgp", según se refiera a llaves públicas o secretas.

Supresión de una llave del llavero

Para suprimir una llave del llavero público:

pgp -kr identificador [llavero]

Este proceso busca en el llavero el identificador indicado y lo suprime si encuentra una coincidencia. Cualquier fragmento del identificador es suficiente para que haya una coincidencia. Se asume que "pubring.pgp" es literalmente el nombre opcional del archivo. Se puede omitir o indicar "secring.pgp" si se quiere suprimir una llave secreta. Se puede dar un nombre distinto para el llavero.

PGP indica el resultado de la operación por pantalla pero puede ser direccionado a un archivo.

Extracción (copia) de una clave del llavero

Para extraer (copiar) una clave del llavero público o secreto:

pgp -kx identificador fdclaves [llavero]

Este proceso copia (sin borrar) la clave especificada por el identificador desde el llavero al archivo indicado ("fdclaves"). El archivo será de tipo binario. Pero si se lo quiere en código ASCII se utiliza las opciones -kxa. El resultado de la operación es informado por PGP y puede ser colocado en un archivo de salida.

Entonces, en determinadas ocasiones el servidor deberá proporcionar a PGP el identificador del propietario e incluso la propia clave pública.

Como PGP envía los resultados de la tarea al Servidor de claves.

Como se explicó ya, en cada operación realizada por PGP, éste produce información explicando el resultado de la misma, y además, cuando se solicita extraer una clave, la graba en un archivo.

Así, el servidor de claves podrá conocer la conclusión de la tarea encomendada a PGP atrapando la salida que esta manda a pantalla y leyendo el archivo donde PGP puso la clave si se le solicitó una.

BIBLIOGRAFIA

1. COMER DOUGLAS E. *Redes globales de información con Internet y TCP/IP: Principios básicos, Protocolos y Aquitectura.* Prentice-Hall Hispanoamericana S.A. (3ra. Edición, México:, 1996); 1-377

2. COMER DOUGLAS E. Y DAVID L. STEVENS, editores *Internetworking with TCP/IP, Volumen III: Client-Server Programming and Applications, Windows Socket Version.* Prentice-Hall Hispanoamericana S.A. (New Jersey: , 1997); 1-488

3. WEBER, *Using JAVA 1.1.* QUE Corporation (3ra. Edición, Indianápolis, 1997).

4. DOUBASALIM, *NetworkingUNIX*, SamsPublishing, Indianapolis-Indiana; 3-396

5. Internet, www.pgp.com

6. Internet, www.Inter_PGP-Home.html