

Escuela Superior Politécnica del Litoral

**FACULTAD DE INGENIERIA EN ELECTRICIDAD
Y COMPUTACION**

“ CORREO POR VOZ ”

**TESIS DE TOPICO
Previa a la Obtención del Título de:
INGENIERO EN COMPUTACION**

Presentado Por:

**MIGUEL ANGEL ALDAZ PACHECO
EFRAIN STYWERT ARGUELLO MORA
RICHARD DE LA TORRE CARVACHE
PAUL GIOVANNY HOYOS VIZCAINO
CARLOS LUIS MENDIETA TRIVIÑO
LOLA KARINA VILLACIS PEÑAHERRERA**

**Guayaquil - Ecuador
1997**

AGRADECIMIENTO

AL ING. JHONNY MACIAS Director de tesis, por su ayuda y colaboración para la realización de este trabajo, a nuestros padres, que con esfuerzo y sacrificio nos impulsaron a terminar esta carrera, y principalmente a Dios por guiarnos por el camino de la vida hacia esta instancia.

DEDICATORIA

A Nuestros Padres Y A Todas
Aquellas Personas Que Nos
Apoyaron De Una U Otra Forma
Para Seguir Avanzando En
Nuestra Vida.



PhD. Enrique Pelaez J.
Miembro del Tribunal



Ing. Guido Caicedo
Miembro del Tribunal



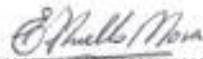
Ing. Jhonny Macias C.
Miembro del Tribunal

DECLARACION EXPRESA

"La responsabilidad por los hechos, ideas y doctrinas expuestos en esta tesis, nos corresponde exclusivamente, y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL"



Miguel Aldaz P.



Stywert Arguello M.

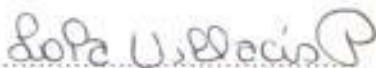
Richard De la Torre C.



Paul Hoyos V.



Carlos Mendieta T.



Lola Villacís P.

RESUMEN

EL proyecto MailVoz, que hemos realizado previa la incorporación de Ingenieros en Computación, lo escogimos por considerar que da un carácter más personal a la comunicación, nos ahorra tiempo, y nos mantiene en contacto con quien estamos comunicándonos.

Los objetivos por nosotros planteados, esperamos que se logren sino en su mayoría en un alto porcentaje, ya que con ello estaremos no solo contribuyendo a mejorar el sistema, mejorando sus funciones y quizás sea un aporte importante al mundo de la tecnología.

Esperamos que este proyecto sirva de ayuda a nuevas generaciones de alumnos que se interesen en mejorar la innovación tecnológica de nuestro tiempo.

El proyecto consta de diversas pantallas de menús, de conexión, de edición, se ha utilizado el Software el Hardware, que son básicos en nuestro proyecto, por lo que tenemos la esperanza que su aplicación sea sumamente útil a futuras generaciones

INDICE GENERAL

RESUMEN	II
INDICE GENERAL	III
INTRODUCCION	1
OBJETIVOS	2
METODOLOGIA	3
REQUERIMIENTOS TECNICOS	4
Capítulo I	6
GENERALIDADES	6
1.1.- ¿QUE ES UNA APLICACIÓN DE CORREO?	6
1.2.- COMPONENTES BASICOS DE UN CORREO	7
1.3.- ¿QUE ES MAPI?	8
Capítulo II	11
ANALISIS Y DISEÑO	11
2.1.- ANALISIS ESTRUCTURAL DE LOS OBJETOS	11
2.1.1.- ESCENARIOS	11
2.1.2.- ESQUEMA GENERAL DE LOS OBJETOS	24
2.1.3.- ESQUEMA DETALLADO DE LOS OBJETOS	25
2.1.4.- DIAGRAMAS DE RAMIFICACION DE LOS OBJETOS	26
2.1.5.- DESCRIPCION DE LOS OBJETOS	27
2.2.- ANALISIS DEL COMPORTAMIENTO DE LOS OBJETOS	29
2.2.1.- ESQUEMA DE EVENTOS	29
2.2.2.- DIAGRAMA DE TRANSICION DE LOS ESTADOS	47
2.3.- DISEÑO DE LA ESTRUCTURA Y COMPORTAMIENTO DE LOS OBJETOS	52
2.4.- DEFINICIÓN DE LAS CLASES	54
2.5.- FLUJO DE VENTANAS	64
2.6.- PLAN DE PRUEBAS DE ACEPTACION	66
2.7.- PROBLEMAS DE IMPLEMENTACION Y ALGORITMOS UTILIZADOS	84
Capítulo III	96
MANUAL DEL USUARIO	96
3.1.- .-INTRODUCCION	96
3.1.1.- CONVENCIONES DEL MANUAL.-	97
3.1.2.- DESCRIPCION DEL AMBIENTE EN QUE SE VA HA TRABAJAR.-	97

3.1.3.- PREGUNTAS SOBRE MAILVOZ:.....	102
3.2.- LO PRIMERO QUE TIENE QUE SABER ES:.....	106
3.2.1.- ¿Cómo conectarse en red?.....	106
3.2.2.- ¿Cuáles son los pasos que se deben seguir para conectarse con el Servidor?.....	106
3.2.3.- ¿Cómo puedo desconectarme del Servidor?.....	109
3.2.4.- ¿Qué es Elegir un Perfil?.....	111
3.2.5.- ¿Se debe conectarse o no con el servidor para reproducir un mensaje que se encuentra en nuestra máquina?.....	112
3.2.6.- ¿Estando en una sesión se puede establecer una nueva?.....	112
3.3.- LO QUE SE DEBE HACER PARA CREAR UN NUEVO MENSAJE.-.....	114
3.3.1.- ¿Cuáles son los pasos que se deben seguir para crear un nuevo mensaje?.....	114
3.3.2.- ¿Qué se debe colocar en la Cabecera del mensaje (Enviar mensaje)?.....	117
3.3.3.- ¿Cómo obtener la dirección de los usuarios que tienen buzones (cuentas) en el Servidor?.....	117
3.3.4.- ¿Se puede comprobar el nombre del usuario a quien yo quiero enviarle el mensaje?.....	120
3.3.5.- ¿Cómo manejo el Reproductor para grabar un mensaje?.....	121
3.3.6.- ¿Cómo se puede adjuntar archivos para ser enviados con el mensaje?.....	122
3.4.- LO QUE SE DEBE HACER PARA LEER UN MENSAJE.-.....	123
3.4.1.- ¿Cuáles son los pasos para Reproducir un Mensaje?.....	123
3.4.2.- ¿Qué nos muestra la Cabecera del mensaje de la pantalla Mensaje (Reproducir un Mensaje)?.....	126
3.4.3.- ¿El Reproductor para reproducir mensajes se maneja igual que el Reproductor para enviar mensajes?.....	127
3.4.4.- ¿Cómo saber si se han enviado archivos adjuntos con el mensaje?.....	128
3.4.5.- ¿Cómo reconocer que un mensaje es nuevo?.....	130
3.4.6.- ¿Cómo buscar los nuevos mensajes que me han llegado?.....	130
3.4.7.- ¿Cómo poder ver los mensajes no leídos?.....	131
3.5.- LO QUE SE DEBE HACER PARA RESPONDER LOS MENSAJES.-.....	132
3.5.1.- Pasos para responder un mensaje.-.....	132
3.5.2.- ¿Cuáles son los pasos para poder enviar un mensaje a todos los usuarios?.....	133
3.6.- LO QUE SE DEBE HACER PARA REENVIAR UN MENSAJE.-.....	135
3.6.1.- ¿Cómo se puede reenviar un mensaje recibido?.....	135
3.7.- LO QUE SE DEBE HACER PARA MOVER MENSAJES.-.....	136
3.7.1.- ¿Cuáles son los pasos para mover los mensajes a carpetas?.....	136
3.8.- LO QUE SE DEBE HACER PARA ELIMINAR UN MENSAJE.-.....	140

3.8.1.- ¿Qué se hace para eliminar un mensaje?.....	140
<i>CONCLUSIONES Y RECOMENDACIONES</i>	142
<i>APENDICE A</i>	144
<i>BIBLIOGRAFIA</i>	156

INTRODUCCION

El propósito de este trabajo es de presentar una nueva forma de utilización del correo, es decir que tenga una aplicación de mayor utilidad ante los avances tecnológicos que impone el nuevo siglo y la modernización del país.

El MailVoz, va a permitir una comunicación mediante la red respectiva con personas, sin que ello implique un gasto extra y más bien estaremos ahorrando tiempo haciendo que haya un diálogo de voz a voz, por cuanto es conocido que no es tan fácil que las personas escriban, en muchas ocasiones no se lo hace por falta de tiempo, o simplemente porque no nos gusta escribir, entonces le estamos dando una nueva opción a todas y cada una de aquellos que viven la era moderna y para quienes el tiempo es factor importantísimo en su agitada vida de empresas o social.

El MailVoz, que es una aplicación de correo por voz, es para nosotros sus creadores una empresa modesta y ambiciosa, modesta en la medida en que está limitada a una investigación y descripción de posibles mejoras en el correo que es un servicio tan útil, pero tan venido a menos en nuestro tiempo y ambiciosa, porque estamos tratando de contribuir con nuestro aporte a una mejor fuente de comunicación, ya que este proyecto tiene una interfaz gráfica amigable, la misma que nos permitirá enviar nuestro mensajes o cartas habladas haciendo uso de nuestra propia habilidad, nuestro sentido del

humor, expresar nuestros sentimientos, con lo que nuestra conversación será más real y en ocasiones más personal.

También queremos recordar que el correo por voz, de hecho es mucho más rápido que el utilizado hasta ahora o sea el escrito, por que de las investigaciones realizadas sabemos que un usuario común puede escribir de 20 a 30 palabras por minuto y los profesionales puede escribir de 60 a 80 palabras por minuto, pero es conocido que podemos hablar de 200 a 250 palabras por minuto, de manera muy confortable y sin cansancio alguno.

OBJETIVOS

Los objetivos que queremos que nuestro sistema cumpla son los siguientes:

- Permitir grabar un mensaje hablado en un archivo.
- Permitir enviar el mensaje dada la dirección del receptor.
- Permitir enviar copias del mensaje original.
- Permitir cancelar el envío de un mensaje.
- Permitir recibir mensajes de otro usuario.
- Permitir reenviar los mensajes de los mensajes recibidos.
- Permitir guardar una copia de los mensajes enviados.
- Permitir almacenar los mensajes enviados y recibidos.
- Permitir eliminar un mensaje tanto recibido como enviado de donde lo almacena el sistema.

- Permitir el acceso a un directorio de usuarios disponibles.
- Permitir guardar los archivos adjuntos en carpetas personales.

METODOLOGIA

Para desarrollar nuestro sistema no basaremos en sistemas de Correo Electrónico ya existentes en el mercado tales como: Microsoft Exchange, Netscape mail, Microsoft Mail, Pegasus, etc., los cuales tienen la capacidad de enviar correo texto enlazado con archivos de cualquier clase, manteniendo la integridad de cada uno, pero formando un solo paquete al ser transmitidos, así mismo tienen la capacidad de recibirlos como un solo paquete y de recuperar cada archivo enviado en el paquete de forma independiente.

El proyecto se lleva a cabo bajo un Análisis y Diseño Orientado a Objetos y como herramienta de trabajo utilizó Visual C++. Se instaló una red con topología Ethernet tipo estrella con protocolo TCP/IP.

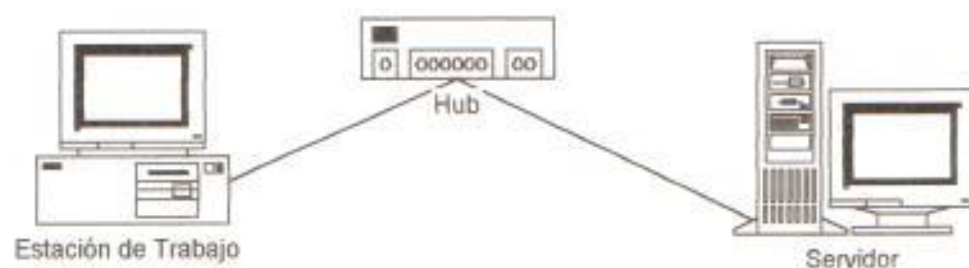


Fig. 1: Red con topología ethernet tipo estrella

Para el desarrollo del sistema se utilizó los Controladores MAPI, el cual puede efectuar una variedad de funciones de Sistema de Mensajería; las cuales nos permiten las siguientes facilidades:

- Acceder a los mensajes actuales en la bandeja de entrada.
- Crear un nuevo mensaje.
- Agregar y eliminar destinatarios de mensajes y datos adjuntos.
- Envía mensajes (con o sin interface de usuario compatible).
- Guardar, copiar y eliminar mensajes.
- Mostrar el cuadro de dialogo **libreta de direcciones**.
- Mostrar el cuadro de dialogo **detalles**.
- Acceder a datos adjuntos, incluyendo los de vinculación e incrustación de objetos (OLE).
- Establecer el nombre de destinatario durante el direccionamiento.
- Efectuar acciones de responder, de responder a todos y de reenviar mensajes.

REQUERIMIENTOS TECNICOS

Los requerimientos de hardware son:

- Una computadora que posee las siguientes características (Servidor):
 - Pentium 166MHZ

- 32MB RAM
- Tarjeta de Red RJ-45
- Disco Duro 1.7GB
- Una computadora que posee las siguientes características (Cliente):
 - Pentium 166MHZ
 - 16MB RAM
 - Tarjeta de Red RJ-45
 - Disco Duro 1.7GB
 - Tarjeta de Sonido
 - Micrófono
- Un concentrador (Hub)
- 2 Cables UTM

Los requerimientos de Software son:

- Windows NT (para el Servidor)
- Windows 95 (para las Estaciones de Trabajo)

Capítulo I

GENERALIDADES

1.1.- ¿QUE ES UNA APLICACIÓN DE CORREO?

Sin lugar a dudas el correo electrónico es el recurso mas utilizado en redes ya que cada día se envía incontable mensajes de una parte a otra de la red, ya que el sistema de correo es un sistema general que puede transportar cualquier tipo de informacion; es decir, que cumpla como unico requisito que sus datos estén almacenados en códigos ASCII. Para entrar mas en detalle, enviar un mensaje (correo electrónico) significa que una computadora debe comunicarse con aquel quien se le envía el mensaje, por tanto debe haber un acuerdo en la forma de comunicarse para esto se usa un estándar de comunicación como es el SMTP (Simple Mail Transfer Protocol) que forma parte de la familia TCP/IP al tener un estándar de comunicación

se garantiza que los mensajes sean dirigidos y transportados de forma correcta.

1.2.- COMPONENTES BASICOS DE UN CORREO

Los componentes básicos que incluye un sistema de correo de voz son como se muestra en la siguiente figura:

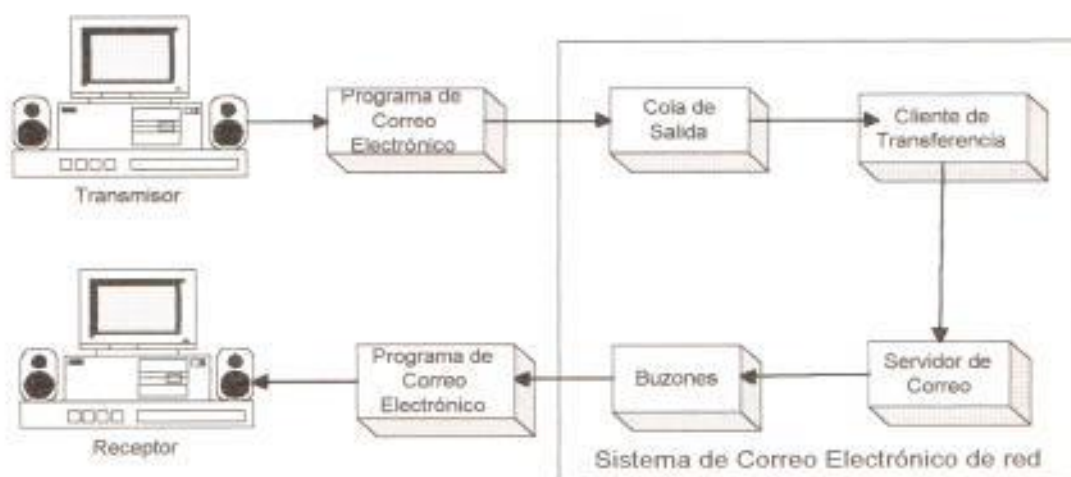


Fig. 2: Elementos básicos de un sistema de correo electrónico de red

Como se observa, tiene los siguientes elementos: Transmisor, Receptor y ambos tienen una interfaz de usuario con el administrador de correo.

El sistema de correo electrónico consta de una cola de salida (colección de mensajes), un proceso cliente, un proceso servidor y buzones para el correo que llega. Aunque la interfaz de usuario (el programa de aplicación de correo electrónico) es, con frecuencia, parte integral del sistema, no es indispensable que lo sea, es decir, puede ser un programa cliente por separado que utiliza el modelo cliente/servidor para interactuar con el sistema de correo. Un **buzón** puede referirse a una dirección de usuario (identificación de usuario y nombre del anfitrión: *usuario@anfitrión.dominio*) o a un **archivo de depósito** que guarda la información de correo electrónico. Imagine el archivo de depósito como una oficina postal del servicio de correo. Al enviar una carta (un mensaje de correo electrónico) al apartado postal de su amigo Pedro (dirección de usuario), la envía de manera física a una oficina postal del servicio de correo (archivo de depósito de correo electrónico). La oficina postal (archivo de depósito) la guarda (correo que llega) hasta que Pedro la recoge.

1.3.- ¿QUE ES MAPI?

Mail Application Programming Interface(MAPI) es un conjunto de funciones que los desarrolladores pueden usar para crear Aplicaciones de Correo. El correo incluye un subconjunto de 12 funciones llamadas "MAPI Simple", que permite a los desarrolladores poder hacer

aplicaciones bajo Windows que permitan enviar, dirigir, y recibir mensajes.

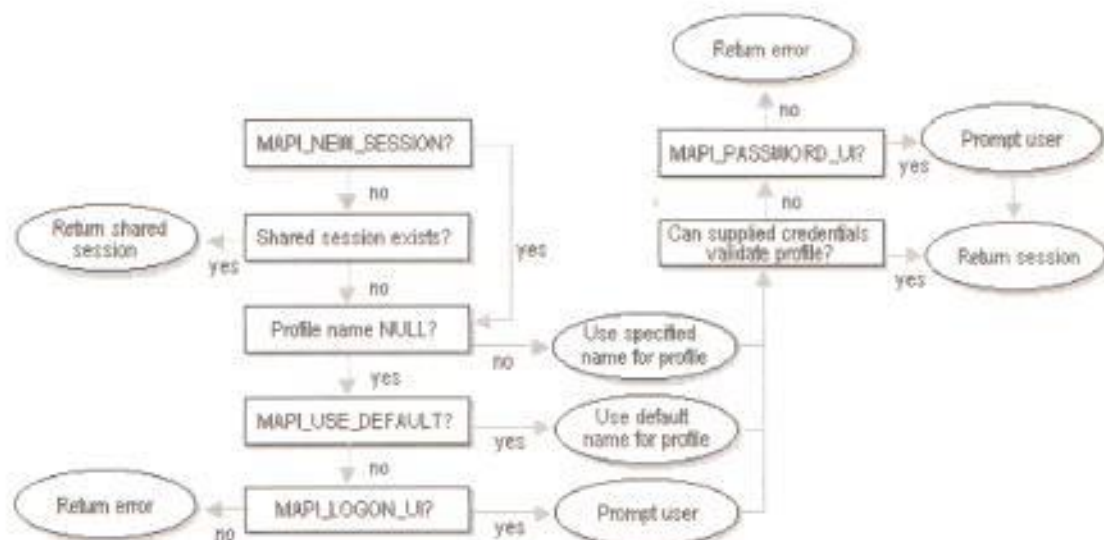


Fig. 3: Como Funciona el MAPI

Con las funciones de "MAPI Simple", los desarrolladores pueden agregar fácilmente el poder de aplicaciones de Mensajerías bajo Windows. MAPI simple apoya la interfaz estándar para la integración simple de aplicaciones de Correo bajo Windows.

Todo las funciones del "MAPI Simple" se diseñan para ser llamadas desde lenguajes de programación como el C o C++, pero ellos también pueden ser llamados desde lenguajes de alto nivel, tal como Visual Basic®, Actor®, Smalltalk®, y ObjectVision®. Las Funciones de "MAPI Simple" pueden también ser llamados desde aplicaciones con

lenguajes de macro que pueden llamar un DLL. Dos ejemplos de estos son Microsoft Excel o Microsoft Word.

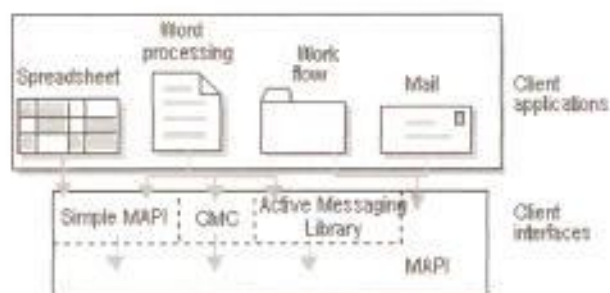


Fig. 4: Integración de las Aplicaciones con MAPI

Enviar los mensajes de correo se controla totalmente desde la aplicación y la biblioteca MAPI de apoyo (MAPI32.DLL).

Los principales aspectos son:

- Una interfaz de programador abierto
- Un propósito dual de interfaz, separado en dos partes independientes
- Una interfaz comprensiva
- La integración con el sistema operativo
- Soporta las normas industriales de desarrollo.

Capítulo II

ANÁLISIS Y DISEÑO

2.1.- ANÁLISIS ESTRUCTURAL DE LOS OBJETOS

2.1.1.- ESCENARIOS

- Caso de uso 1: Conectarse.
 - ◆ Escenario 1.1: Inicio de sesión
 - Suposiciones:
 - ✓ No hay sesión establecida.
 - ✓ La lista de mensajes está vacía.
 - ✓ No hay error en la conexión.
 - Resultados:
 - ✓ Se establece la sesión.
 - ✓ El control de mensajes accesa al servidor.

- ✓ El control de mensajes transmite mensajes a la bandeja de entrada.
- ✓ La lista de mensajes es actualizada con la bandeja de entrada.

◆ Escenario 1.2: Establecimiento de nueva sesión

• Suposiciones:

- ✓ Existe una sesión.
- ✓ No hay error en la conexión.

• Resultados:

- ✓ Se termina la sesión anterior y se establece una nueva.
- ✓ Se anula la lista de mensajes.
- ✓ El control de mensajes accesa al servidor.
- ✓ El control de mensajes transmite mensajes a la bandeja de entrada.
- ✓ La lista de mensajes es actualizada con la bandeja de entrada.

□ Caso de uso 2: Reproducir un mensaje.

◆ Escenario 2.1: Reproducción de un mensaje de audio.

• Suposiciones:

- ✓ El cuerpo de mensaje es de audio.

- ✓ Hay un mensaje seleccionado.
- ✓ No hay errores en el archivo de audio.
- Resultados:
 - ✓ El reproductor abre el cuerpo del mensaje y está listo para reproducirlo.
 - ✓ La cabecera del mensaje y los recipientes son cargados.
 - ✓ Los adjuntos son cargados en una lista.

- ◆ Escenario 2.2: Reproducción de un mensaje de otro formato.
 - Suposiciones:
 - ✓ El cuerpo de mensaje no es de audio.
 - ✓ Hay mensaje seleccionado.
 - Resultados:
 - ✓ El reproductor está inhabilitado.
 - ✓ La cabecera del mensaje y los recipientes son cargados.
 - ✓ Los adjuntos son cargados en una lista.

- Caso de uso 3: Guardar un adjunto.
 - ◆ Escenario 3.1: Guardando un adjunto.
 - Suposiciones:
 - ✓ Hay un mensaje en reproducción.

- ✓ El mensaje tiene adjuntos.
- ✓ Hay un adjunto seleccionado.
- ✓ Se define una trayectoria válida.
- Resultados:
 - ✓ El adjunto es guardado en el archivo especificado en la trayectoria.

- Caso de uso 4: Crear nuevo mensaje.
 - ◆ Escenario 4.1: Nuevo mensaje.
 - Suposiciones:
 - ✓ Hay una sesión establecida.
 - Resultados:
 - ✓ El control de mensaje es seteado en modo de escritura.
 - ✓ El reproductor es abierto en modo de grabación.
 - ✓ Se habilita opción de adjuntar.
 - ✓ Se habilita opción de ver libreta de direcciones

- Caso de uso 5: Adjuntar archivo.
 - ◆ Escenario 5.1: Se adjunta archivo a nuevo mensaje.
 - Suposiciones:
 - ✓ El control de mensaje está seteado en modo de escritura.

- ✓ El archivo existe.
- Resultados:
 - ✓ Se inserta un adjunto al nuevo mensaje.
 - ✓ Se incrementa el contador de adjuntos.

- Caso de uso 6: Escoger recipientes de la libreta de direcciones.
 - ◆ Escenario 6.1: Cargar la lista de recipientes del nuevo mensaje.
 - Suposiciones:
 - ✓ La libreta de direcciones está disponible.
 - ✓ El control de mensajes está seteado en escritura.
 - Resultados:
 - ✓ Se inserta los recipientes escogidos en la libreta de direcciones al nuevo mensaje.

- Caso de uso 7: Enviar nuevo mensaje.
 - ◆ Escenario 7.1: Enviando nuevo mensaje.
 - Suposiciones:
 - ✓ El cuerpo del mensaje ha sido creado.
 - ✓ Se ha definido recipientes.
 - Resultados:
 - ✓ El control de mensaje transmite el nuevo mensaje.

- Caso de uso 8: Responder un mensaje.
 - ◆ Escenario 8.1: Respondiendo un mensaje.
 - Suposiciones:
 - ✓ Hay un mensaje seleccionado.
 - ✓ Se ha establecido una sesión.
 - Resultados:
 - ✓ Se setea el control de mensaje en modo de escritura.
 - ✓ Se setea el recipiente destino con el recipiente remitente del mensaje seleccionado.
 - ✓ Se abre el reproductor en modo de grabación.
 - ✓ Se habilita opción de adjuntar.
 - ✓ Se habilita opción de ver libreta de direcciones.

- Caso de uso 9: Responder a todos.
 - ◆ Escenario 9.1: Respondiendo un mensaje a todos.
 - Suposiciones:
 - ✓ Hay un mensaje seleccionado.
 - ✓ Se ha establecido una sesión.
 - Resultados:
 - ✓ Se setea el control de mensaje en modo de escritura.
 - ✓ Se setea los recipientes del mensaje seleccionado al mensaje nuevo.
 - ✓ Se abre el reproductor en modo de grabación.

- ✓ Se habilita opción de adjuntar .
- ✓ Se habilita opción de ver libreta de direcciones.

□ Caso de uso 10: Reenviar un mensaje.

◆ Escenario10.1:Reenviando un mensaje.

• Suposiciones:

- ✓ Hay un mensaje seleccionado.
- ✓ Se ha establecido una sesión.

• Resultados:

- ✓ Se setea el control de mensaje en modo de escritura.
- ✓ Se copia el cuerpo del mensaje seleccionado al mensaje nuevo.
- ✓ Se abre el reproductor en modo de reproducción.
- ✓ Se inhabilita opción de adjuntar.
- ✓ Se habilita opción de ver libreta de direcciones.

□ Caso de uso 11: Buscar correo nuevo.

◆ Escenario11.1:Buscando correo nuevo.

• Suposiciones:

- ✓ Se ha establecido una sesión.

• Resultados:

- ✓ Se anula la lista de mensajes.
- ✓ El control de mensajes accesa al servidor.

- ✓ El control de mensajes transmite mensajes a la bandeja de entrada.
 - ✓ La lista de mensajes es actualizada con la bandeja de entrada.
 - ✓ Se cuentan los mensajes.
- Caso de uso 12: Ver mensajes no leídos.
- ◆ Escenario 12.1: Ver mensajes no leídos de la bandeja de entrada.
 - Suposiciones:
 - ✓ Se ha establecido una sesión.
 - ✓ Están cargados los mensajes en la bandeja de entrada.
 - Resultados:
 - ✓ Se anula la lista de mensajes.
 - ✓ El control de mensajes accesa al servidor.
 - ✓ El control de mensajes transmite solo los mensajes no leídos a la bandeja de entrada.
 - ✓ La lista de mensajes es actualizada con la bandeja de entrada.
 - ✓ Se cuentan los mensajes.

- ◆ Escenario12.2: Ver mensajes no leídos de una carpeta.
 - Suposiciones:
 - ✓ Están cargados los mensajes de una carpeta.
 - Resultados:
 - ✓ Se anula la lista de mensajes.
 - ✓ La lista de mensajes es actualizada solo con los mensajes no leídos de la carpeta actual.
 - ✓ Se cuentan los mensajes.

- Caso de uso 13: Ver todos mensajes.
 - ◆ Escenario13.1: Ver todos los mensajes de la bandeja de entrada.
 - Suposiciones:
 - ✓ Se ha establecido una sesión.
 - ✓ Están cargados los mensajes en la bandeja de entrada.
 - Resultados:
 - ✓ Se anula la lista de mensajes.
 - ✓ El control de mensajes accesa al servidor.
 - ✓ El control de mensajes transmite todos los mensajes a la bandeja de entrada.
 - ✓ La lista de mensajes es actualizada con la bandeja de entrada.

- ✓ Se cuentan los mensajes.

- ◆ Escenario13.2: Ver todos los mensajes de una carpeta.
 - Suposiciones:
 - ✓ Están cargados los mensajes de una carpeta.
 - Resultados:
 - ✓ Se anula la lista de mensajes.
 - ✓ La lista de mensajes es actualizada todos con los mensajes de la carpeta actual.
 - ✓ Se cuentan los mensajes.

- Caso de uso 14: Mover.
 - ◆ Escenario14.1: Mover mensajes a una carpeta existente.
 - Suposiciones:
 - ✓ Se ha seleccionado un mensaje.
 - ✓ La carpeta existe.
 - Resultados:
 - ✓ Se copia el mensaje a la carpeta seleccionada.
 - ✓ Se elimina el mensaje de la lista.
 - ✓ Se guarda la carpeta en su archivo.

◆ Escenario14.2:Mover mensajes a una carpeta nueva.

• Suposiciones:

- ✓ Se ha seleccionado un mensaje.
- ✓ La carpeta no existe.

• Resultados:

- ✓ Se crea la carpeta y el archivo relacionado.
- ✓ Se copia el mensaje a la carpeta nueva.
- ✓ Se elimina el mensaje de la lista.
- ✓ Se guarda la carpeta en su archivo.
- ✓ Se actualiza la lista de carpetas.

□ Caso de uso 15: Eliminar.

◆ Escenario15.1:Eliminar un mensaje de una carpeta.

• Suposiciones:

- ✓ Se ha seleccionado un mensaje.

• Resultados:

- ✓ Se copia el mensaje a la carpeta basurero.
- ✓ Se elimina el mensaje de la lista.
- ✓ Se guarda la carpeta en su archivo.

- ◆ Escenario15.2: Eliminar un mensaje de la bandeja de entrada.

- Suposiciones:

- ✓ Se ha seleccionado un mensaje.

- ✓ Existe una sesión establecida.

- Resultados:

- ✓ Se copia el mensaje a la carpeta basurero.

- ✓ Se elimina el mensaje de la bandeja de entrada.

- ✓ El control de Mensaje elimina el mensaje del servidor.

- ◆ Escenario15.3: Eliminar un mensaje del basurero.

- Suposiciones:

- ✓ Es abierta la carpeta basurero.

- ✓ Se ha seleccionado un mensaje.

- Resultados:

- ✓ Se elimina el mensaje de la lista.

- ✓ Se guarda el basurero en su archivo.

- ✓ Se actualiza la lista.

- Caso de uso 16: Salir.

- ◆ Escenario16.1: Salir y vaciar basurero.

- Suposiciones:

- ✓ El basurero no esta vacío.

- Resultados:
 - ✓ Se guarda el basurero vacío a su archivo.
 - ✓ Se guarda la carpeta actual
 - ✓ Se cancela la sesión.

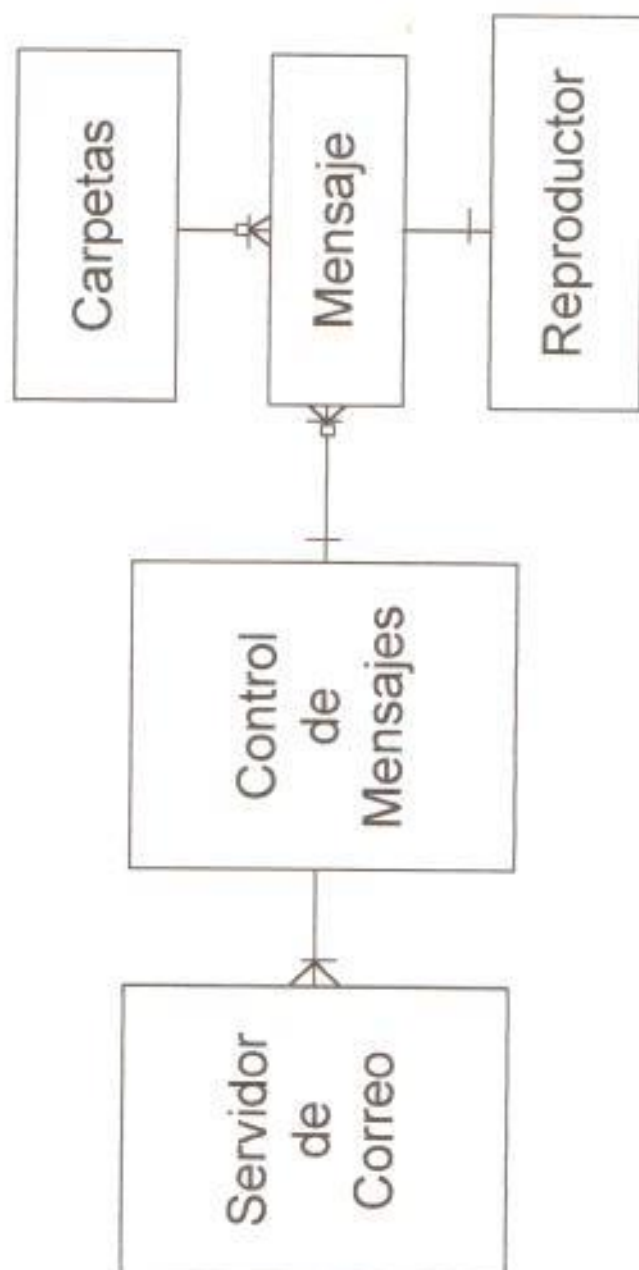


Fig. 5 : Esquema General de Objetos

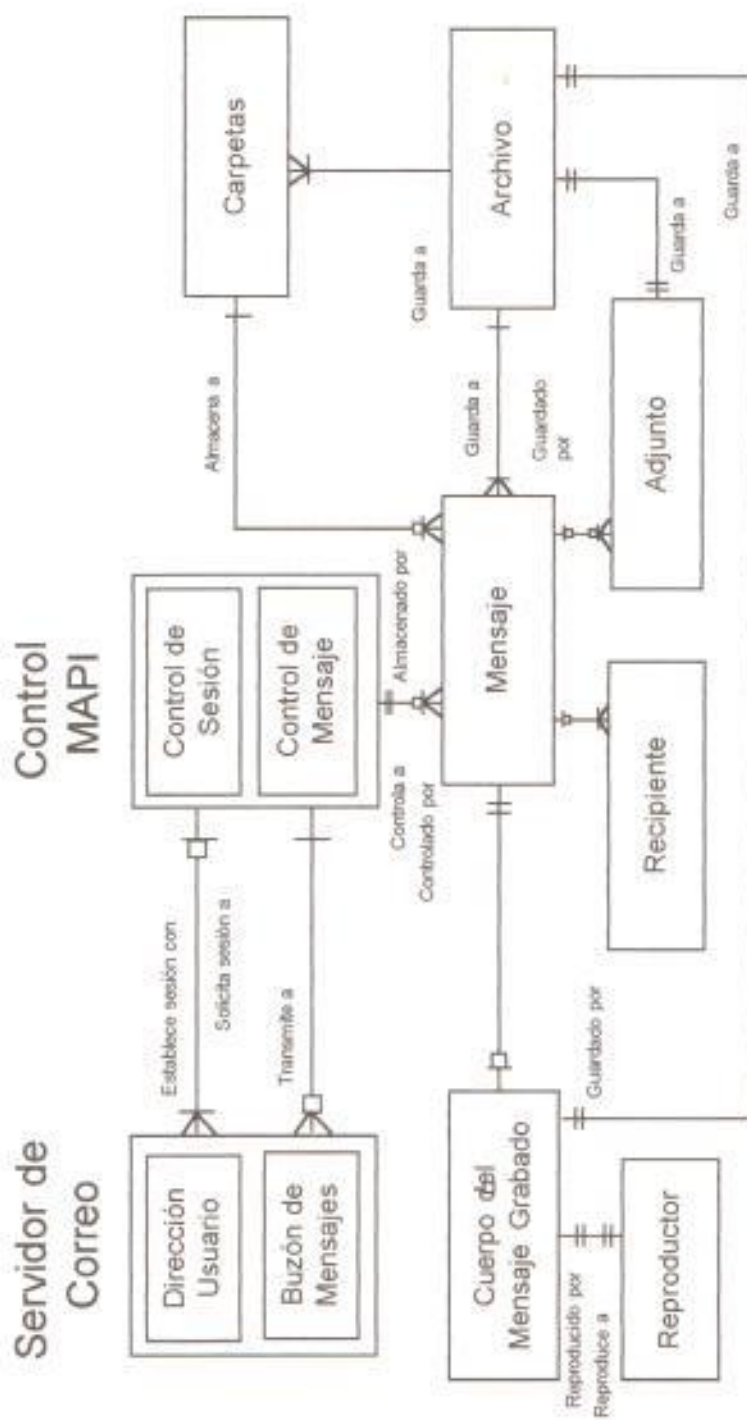


Fig. 6 : Esquema Detallado de Objetos

2.1.4.- DIAGRAMAS DE RAMIFICACION DE LOS OBJETOS

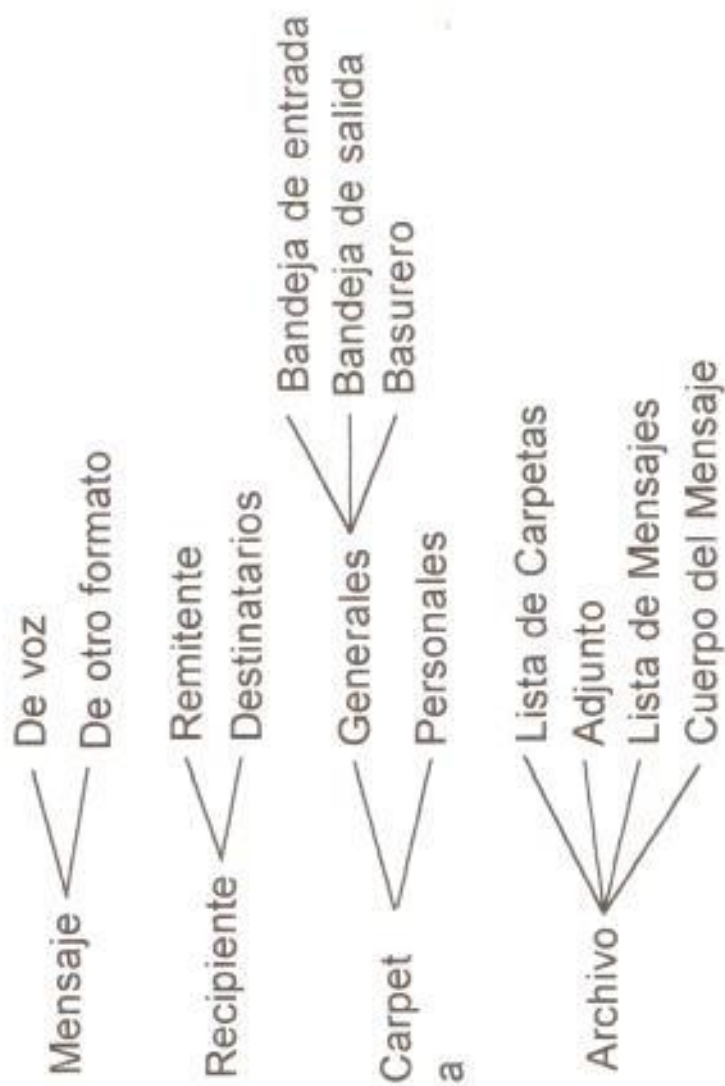


Fig. 7 : Diagrama de Ramificación

2.1.5.-DESCRIPCION DE LOS OBJETOS

- **Control de sesión.**- Es el encargado de establecer el enlace entre la aplicación cliente y el servidor basándose en un usuario y contraseña, para tener acceso a los mensajes almacenados en el respectivo buzón del cliente que ha realizado la sesión.
- **Control de mensaje.**- Se encarga de la transmisión y recepción de mensajes a y desde el servidor, basándose en el identificador de sesión conseguido por el Control de Sesión. También brinda la facilidad de acceder a la Libreta de Direcciones.
- **Servidor de Correo.**- Este es un objeto externo al Sistema con el cual se establecerá la comunicación para la transmisión de mensajes. Válida la contraseña del usuario y establece la comunicación ante una solicitud del Control de Sesión, proveyendo el identificador de sesión. Contiene buzones asignados a los usuarios con sus respectivos mensajes.
- **Recipiente.**- Son el remitente y los destinatarios originales y con copias del mensaje. Estos tienen una dirección física que permite llegar a sus buzones respectivos dentro del servidor de correo.

- Adjunto.- Son archivos que se pueden enlazar con el mensaje para luego ser extraídos y guardado en disco. Estos archivos pueden ser de varios tipos, y tienen un nombre y trayectoria o path.
- Cuerpo del Mensaje grabado.- Es el archivo que contiene el mensaje grabado por el usuario. Es de formato wav con alta calidad y podría ser teóricamente de cualquier tamaño.
- Mensaje.- Es el objeto central del Sistema. Está compuesto de la lista de Recipientes, la lista de Adjuntos, y Cuerpo del Mensaje.
- Carpeta.- Existen dos clases de carpetas: Generales y Personales. Las primeras son la bandeja de Entrada, de Salida y el Basurero. Las segundas le permiten al usuario organizar sus mensajes en carpetas personalizadas.
- Bandeja de Entrada.- Es la carpeta en la que están contenidos todos los mensajes que le han llegado al usuario.
- Bandeja de Salida.- Es la carpeta que guarda los mensajes que el usuario ha enviado.
- Basurero.- Esta carpeta guarda los mensajes eliminados para que el usuario pueda recuperarlos hasta que los elimine de la misma.
- Reproductor.- Es el objeto encargado de brindar las facilidades de reproducción y grabación de mensajes.

2.2.- ANALISIS DEL COMPORTAMIENTO DE LOS OBJETOS

2.2.1.- ESQUEMA DE EVENTOS

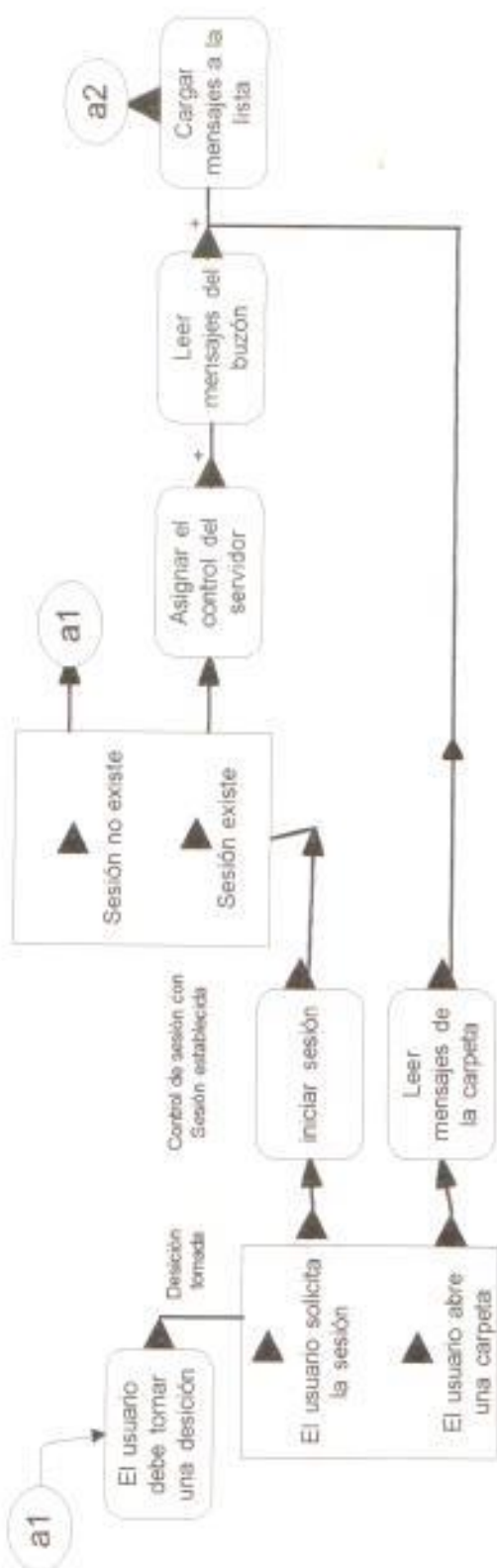


Fig. 8 : Esquema de los Eventos. Sección 1

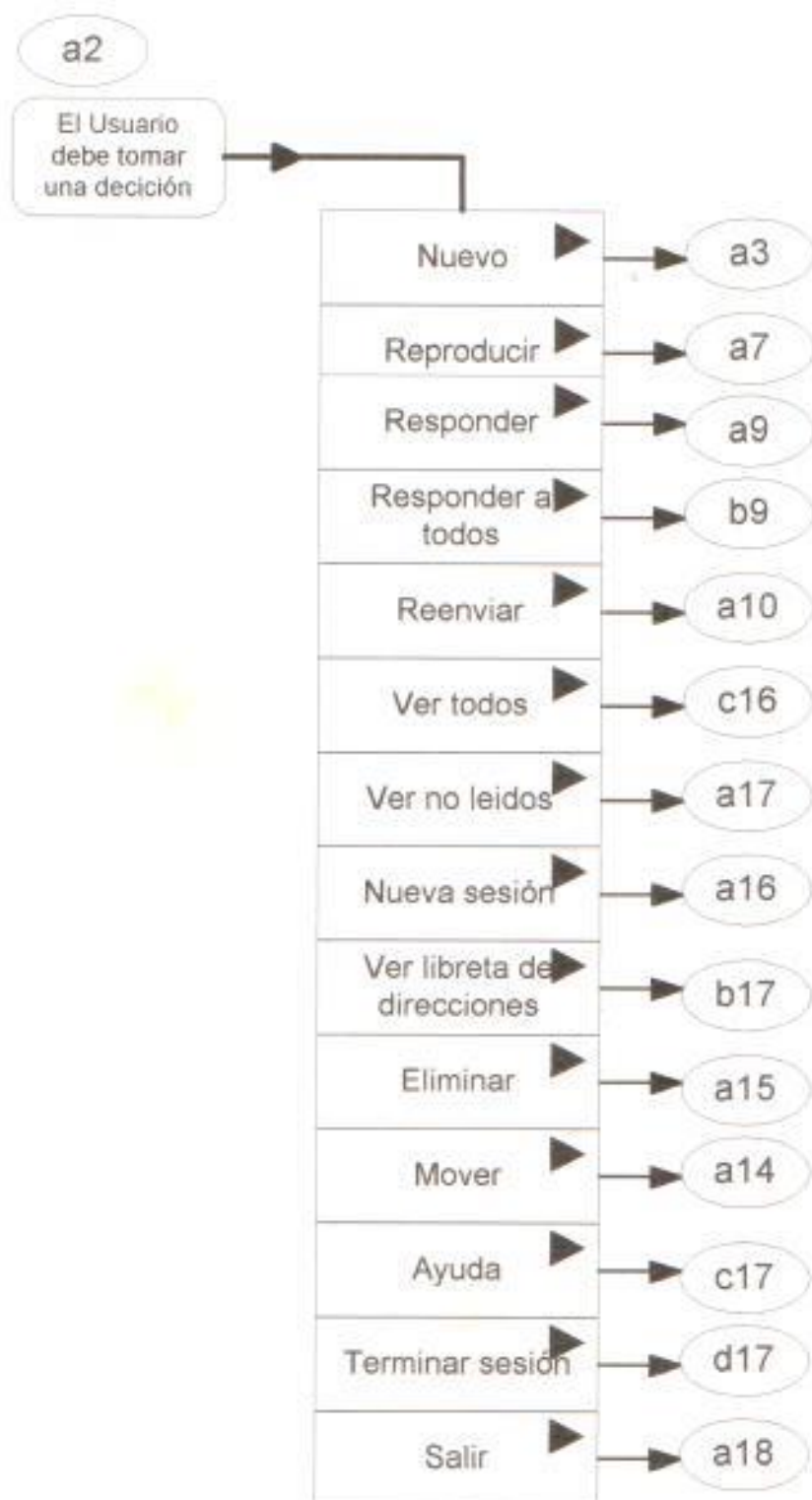


Fig. 9 : Esquema de Eventos Sección 2

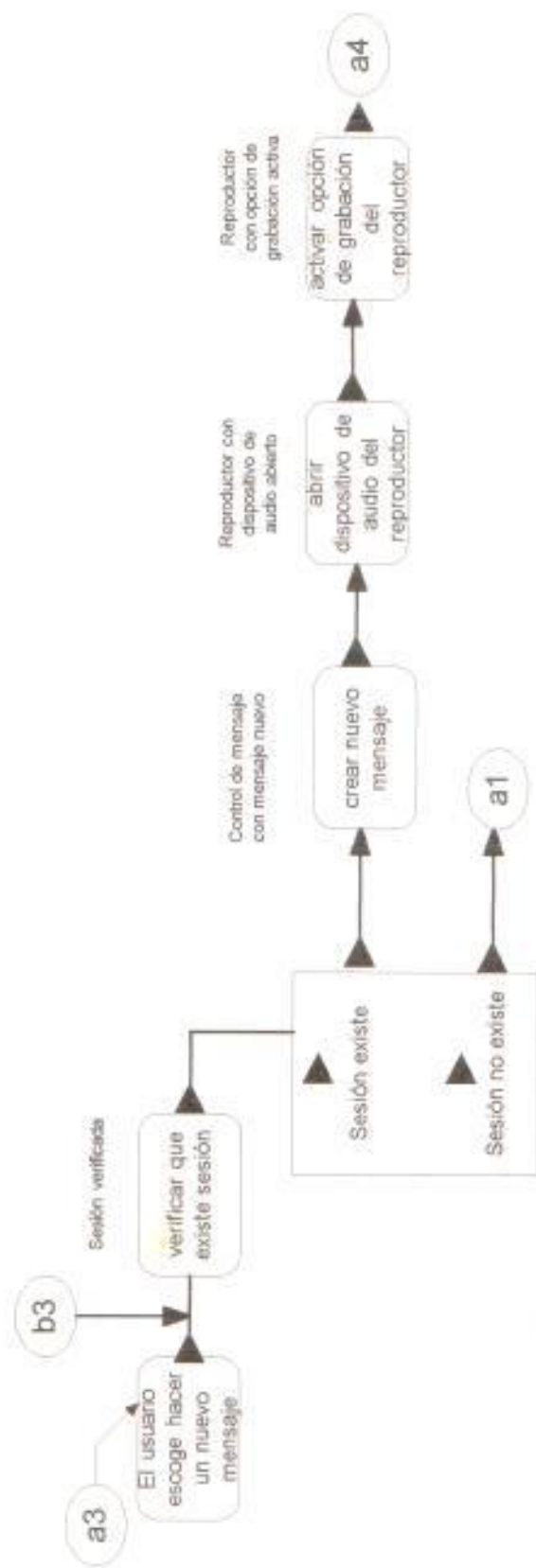


Fig. 10.: Esquemas de Eventos Sección 3

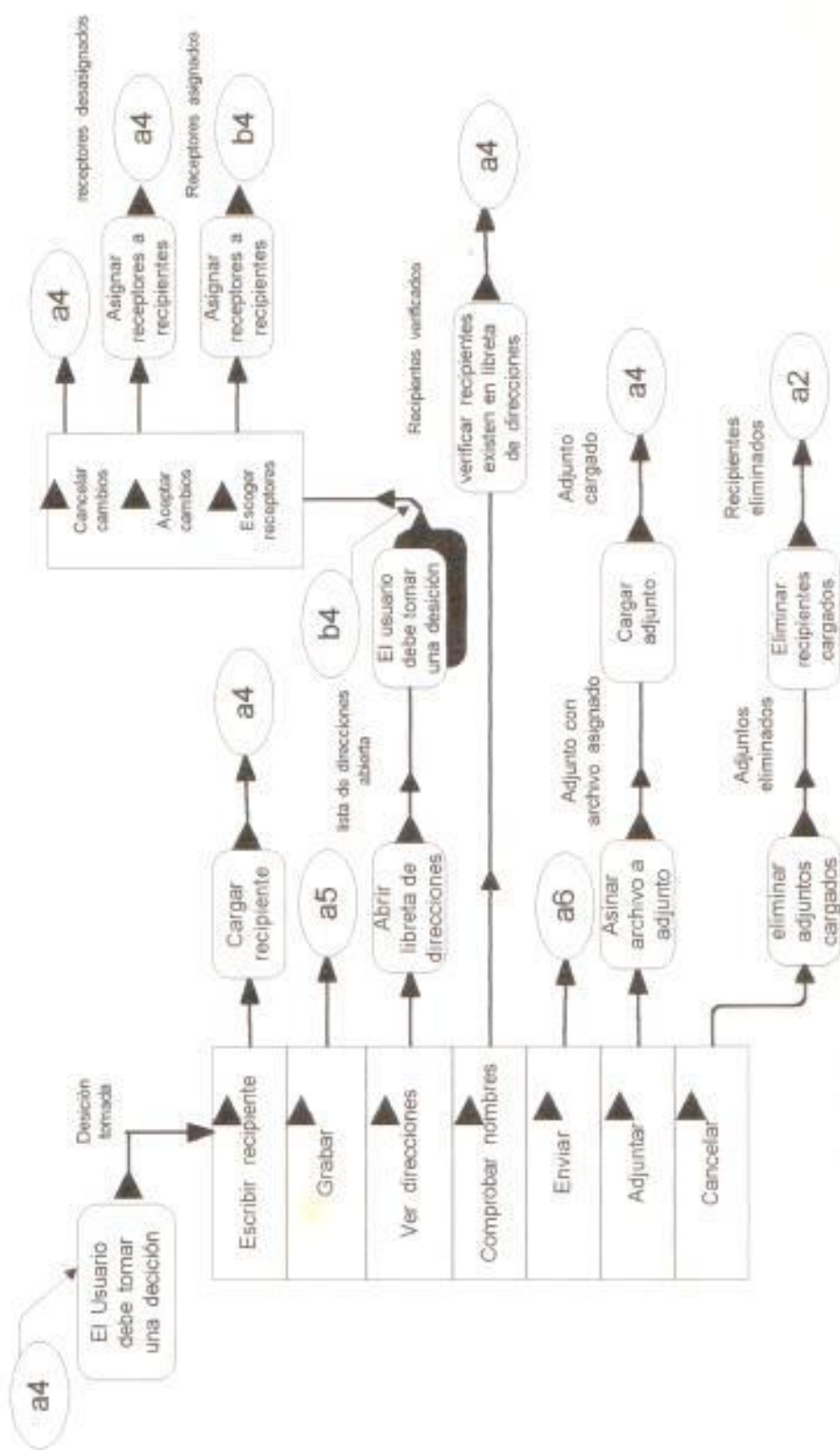


Fig. 11 : Esquema de Eventos Sección 4

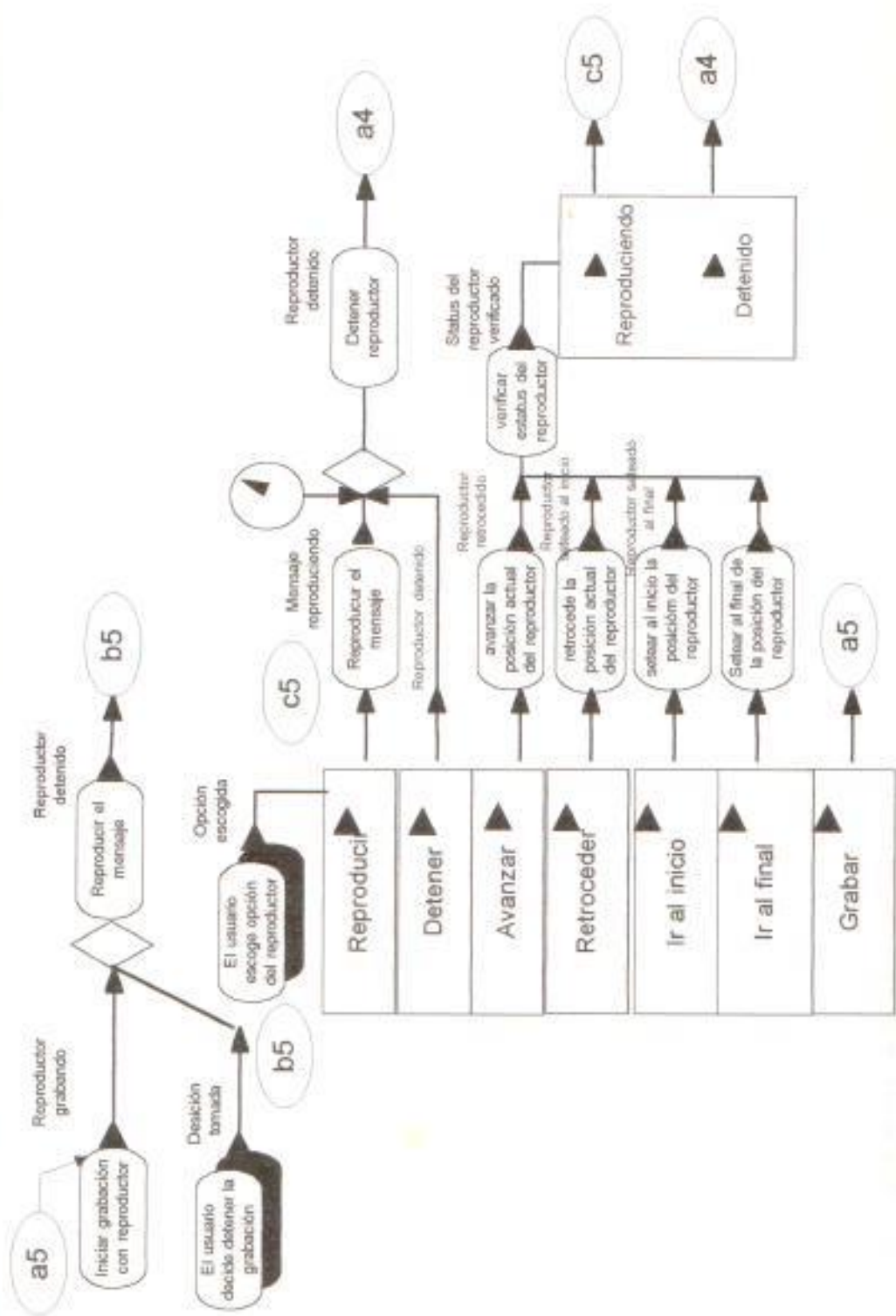


Fig. 12 : Esquemas de Eventos. Sección 5

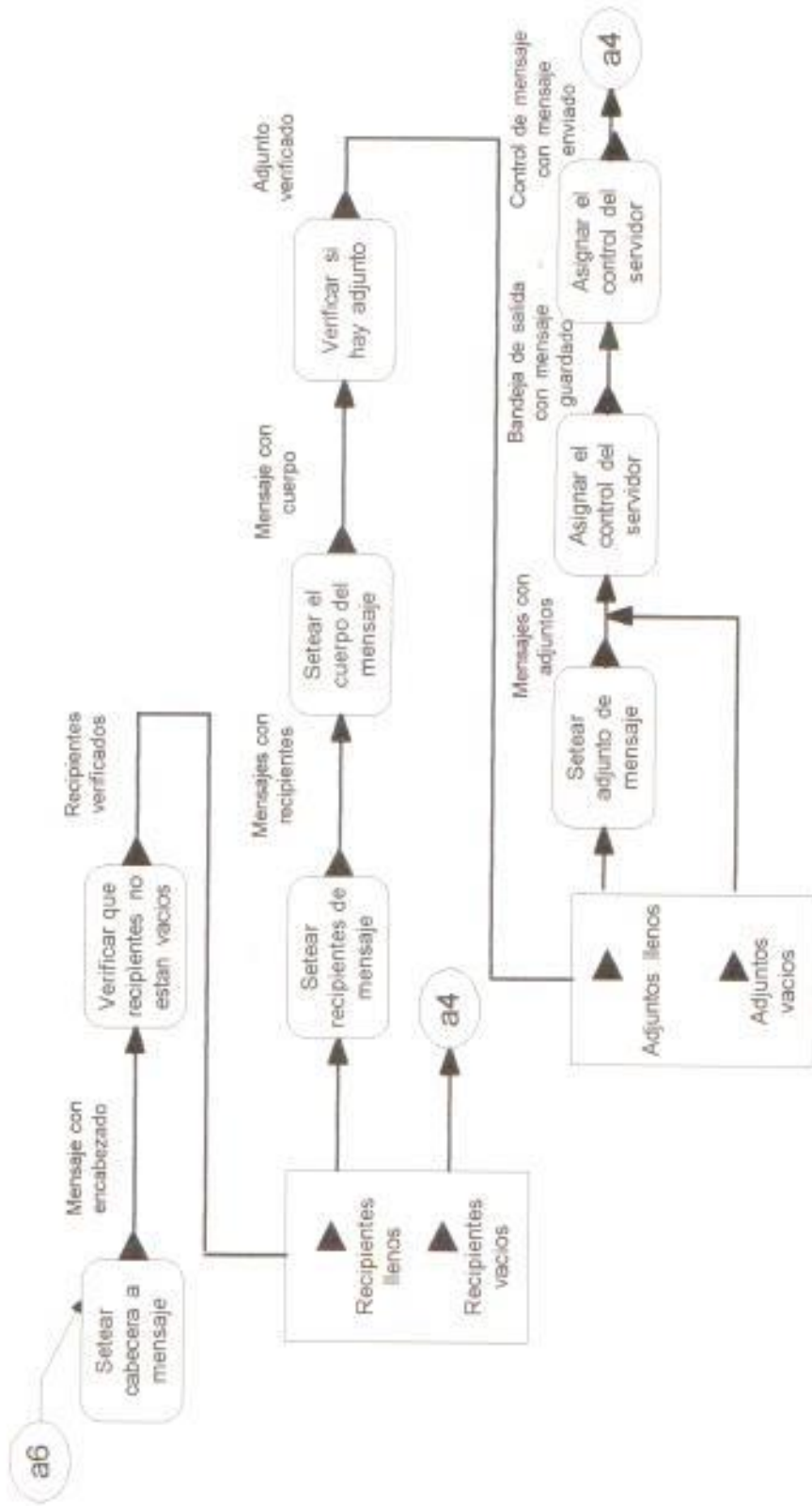


Fig. 13 : Esquemas de Eventos. Sección 6

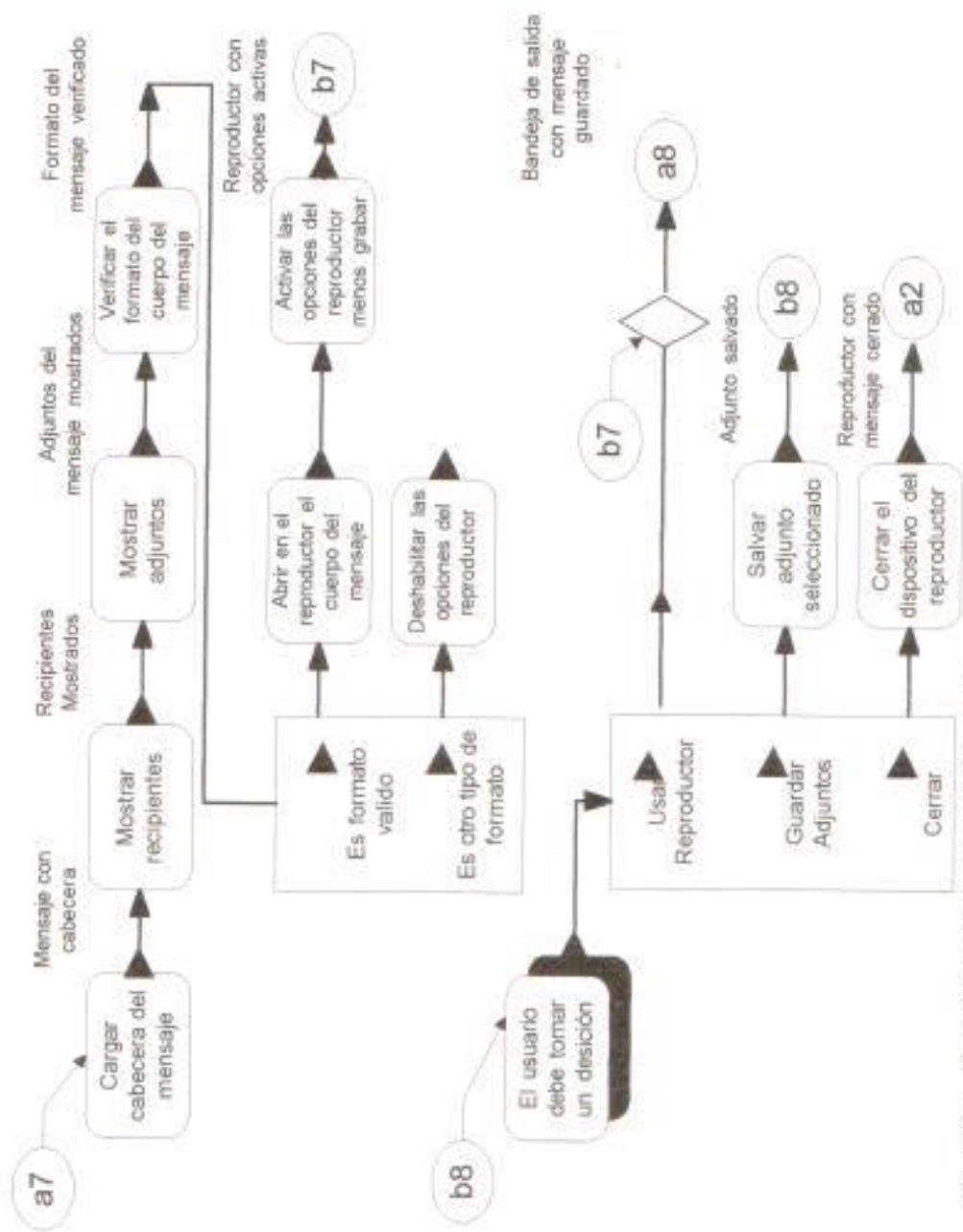


Fig. 14 : Esquemas de Eventos Sección 7

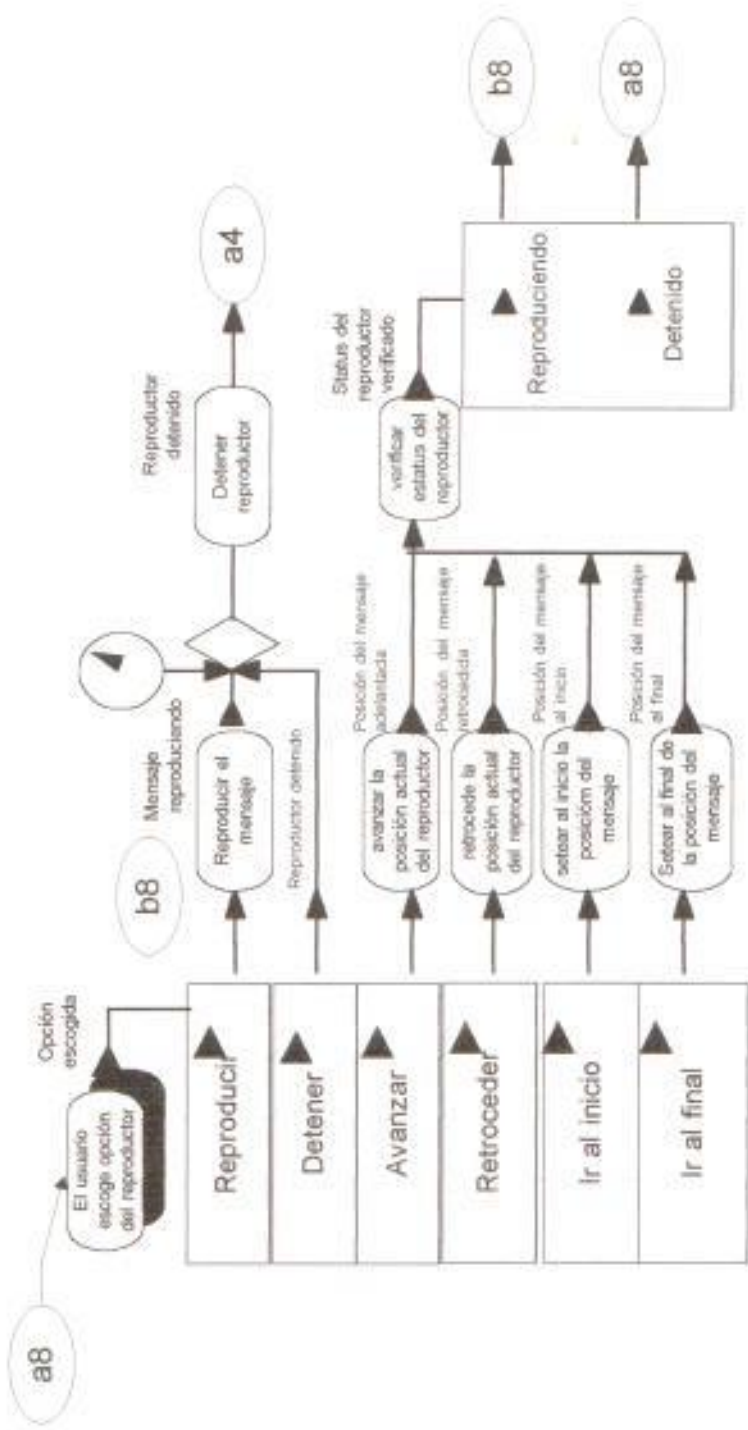


Fig. 15 : Esquemas de Eventos. Sección 8

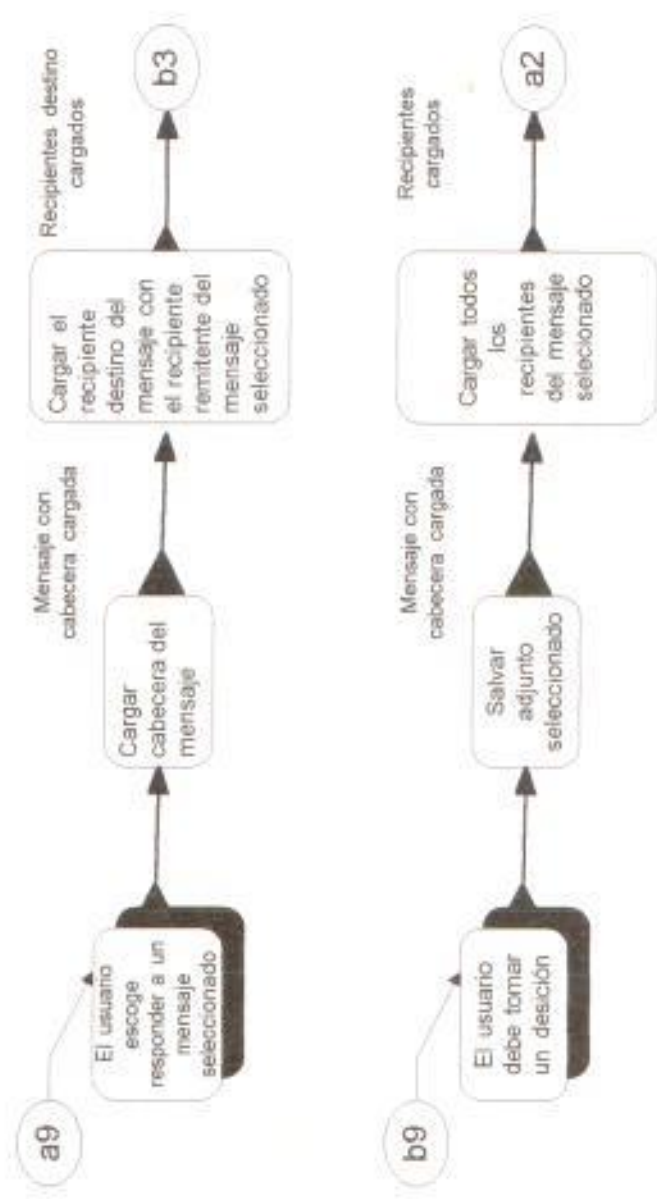


Fig. 16 : Esquemas de Eventos. Sección 9

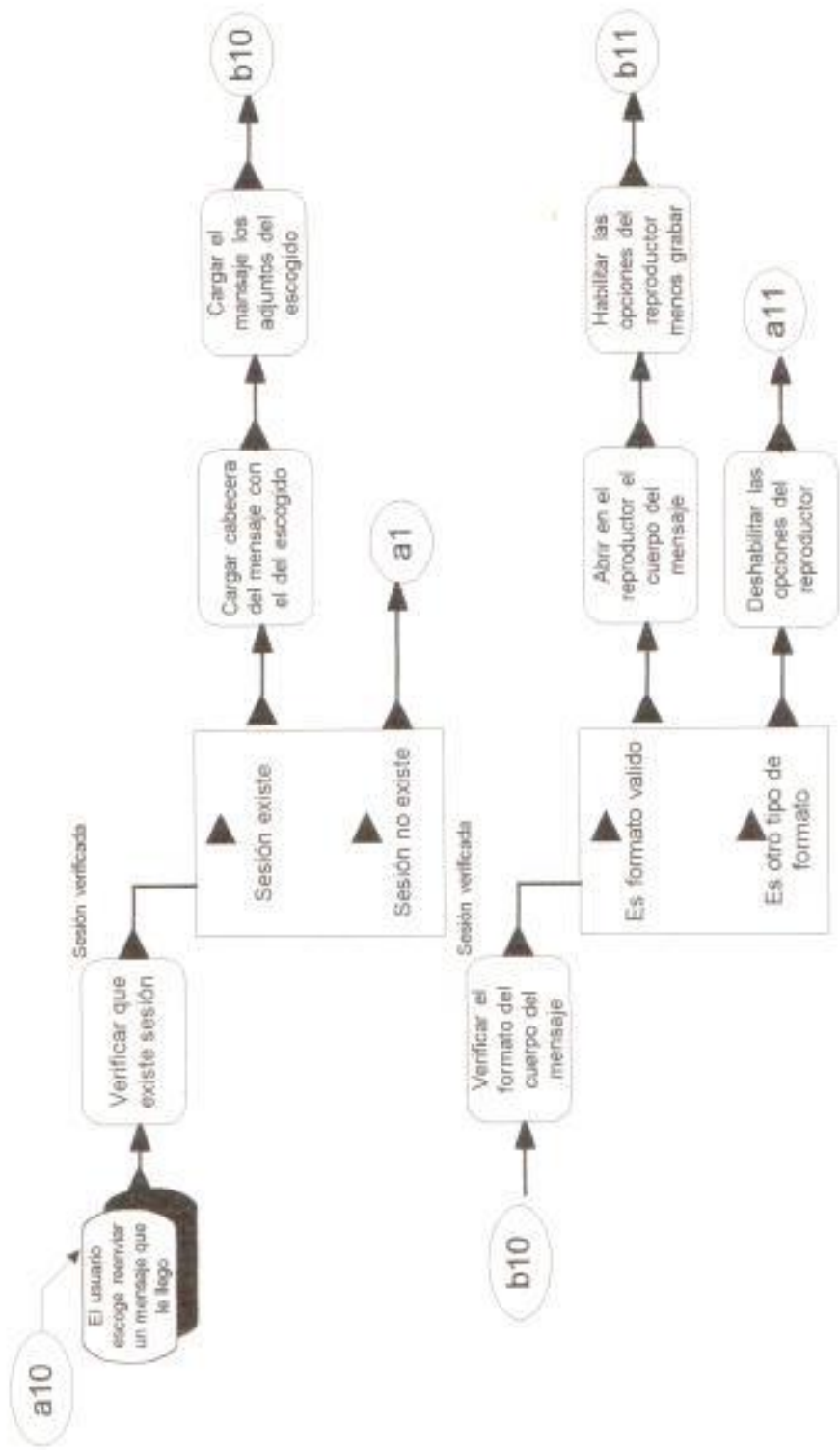


Fig. 17 : Esquemas de Eventos Sección 10

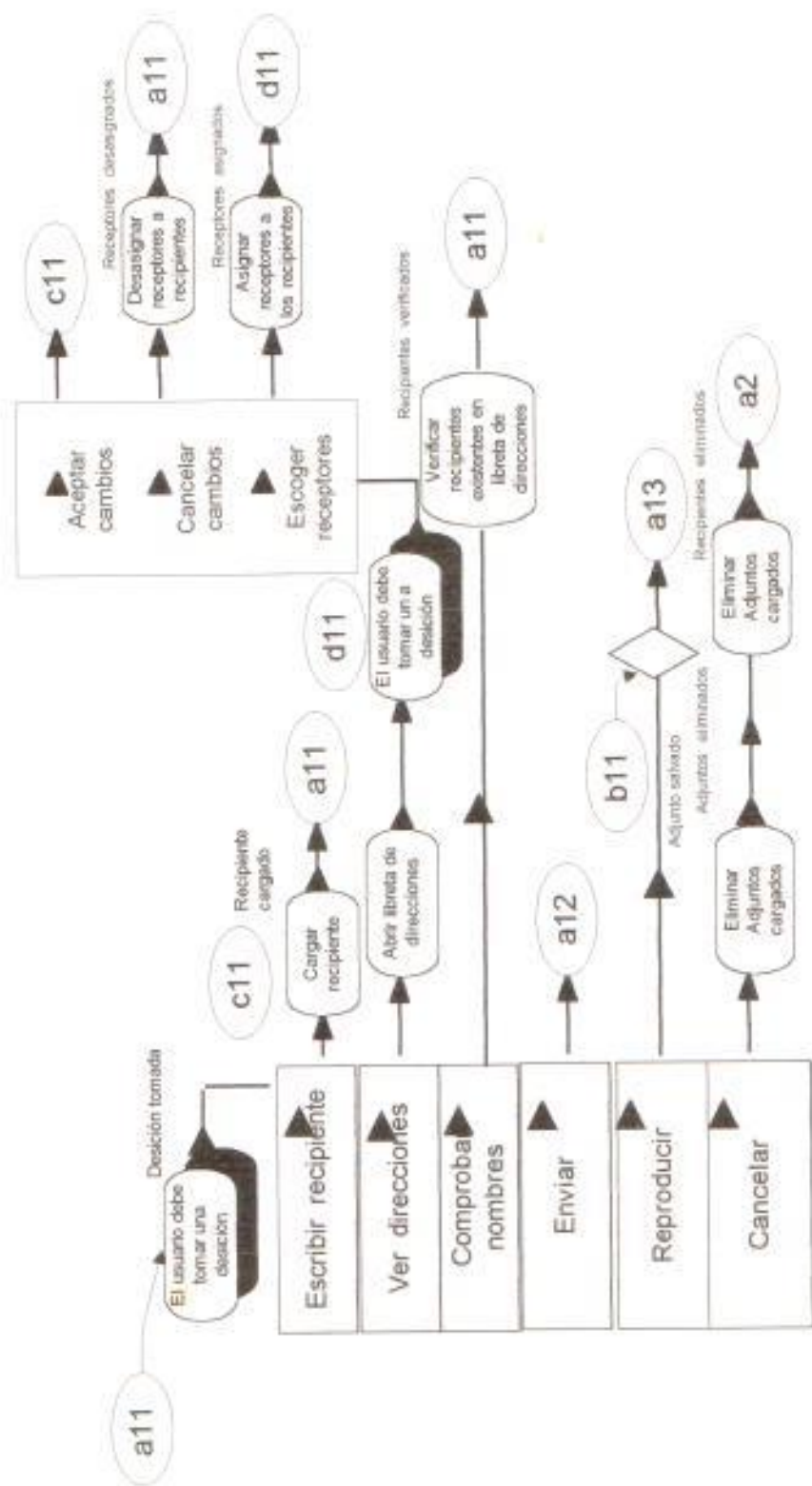


Fig. 18: Esquemas de Eventos. Sección 11

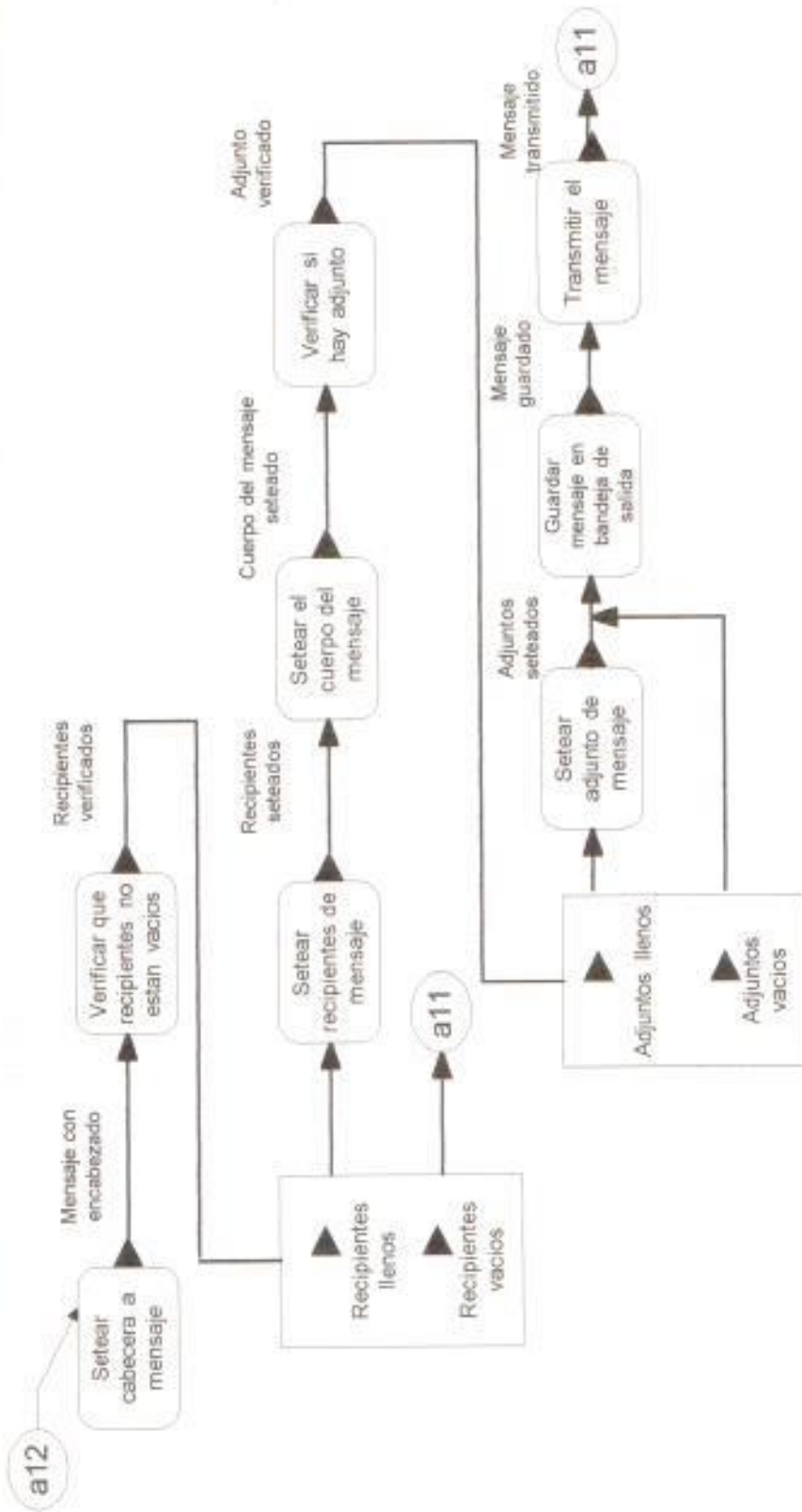


Fig. 19 : Esquemas de Eventos. Sección 12

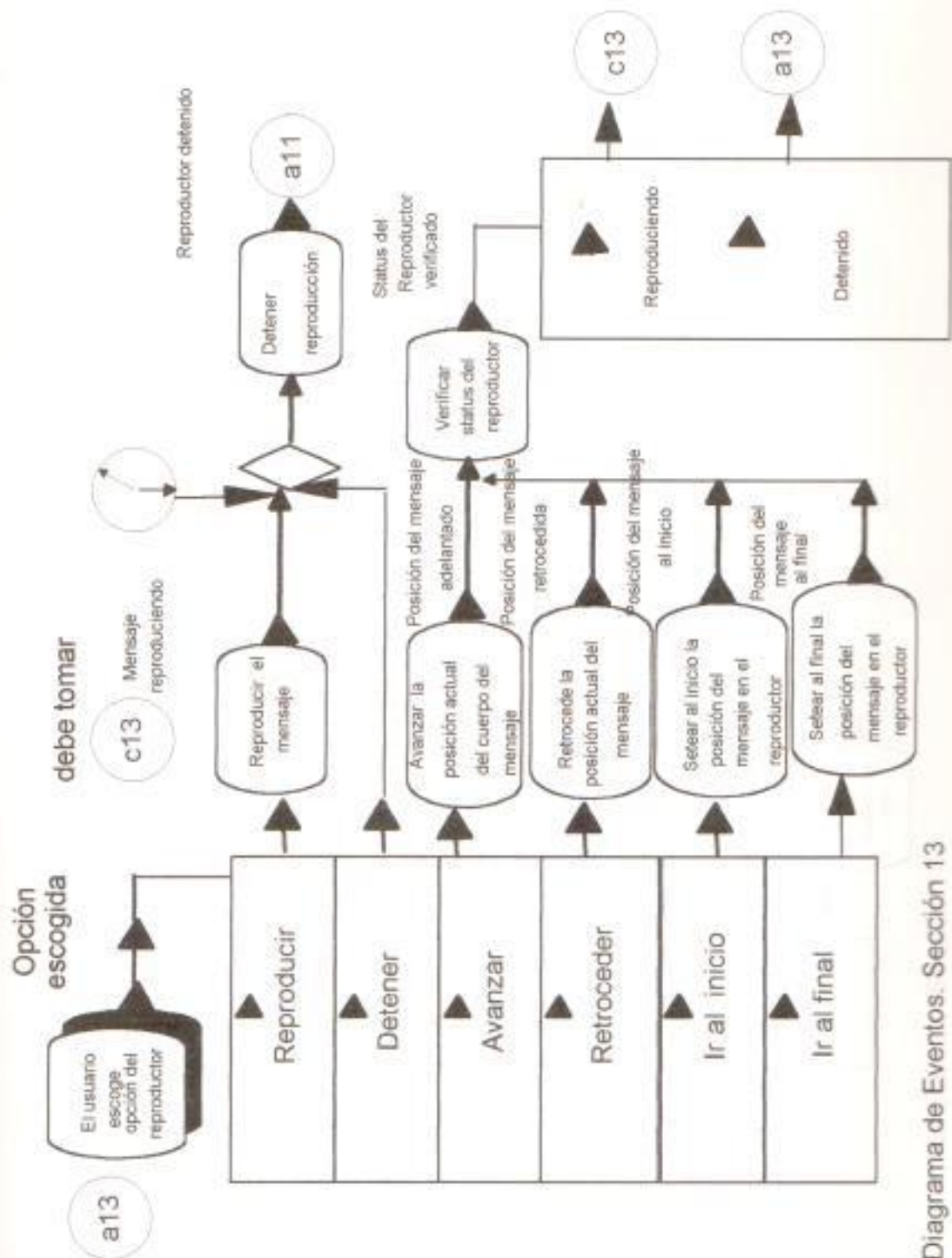


Fig.20 Diagrama de Eventos. Sección 13

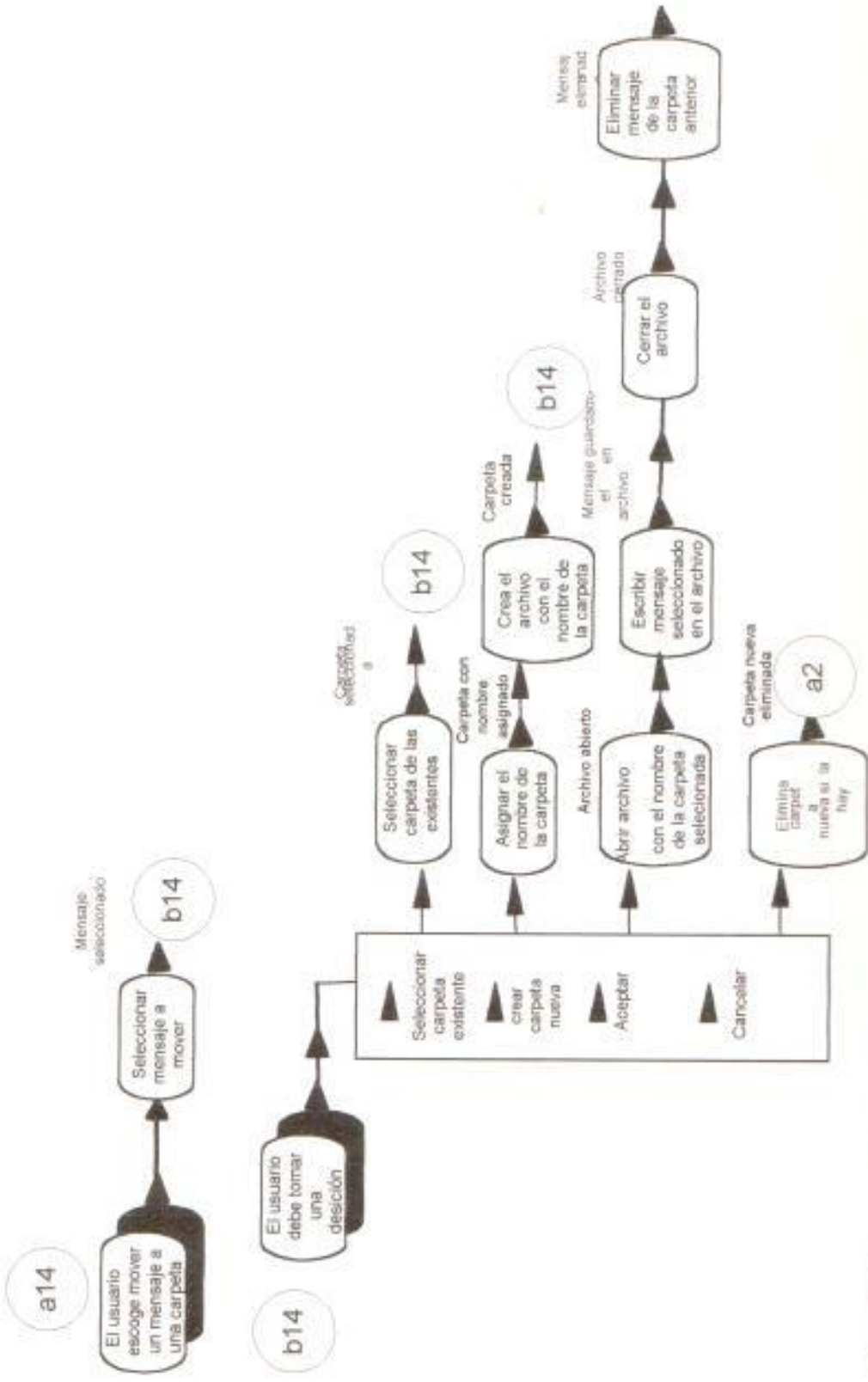


Fig. 21 Diagrama de Eventos. Sección 14

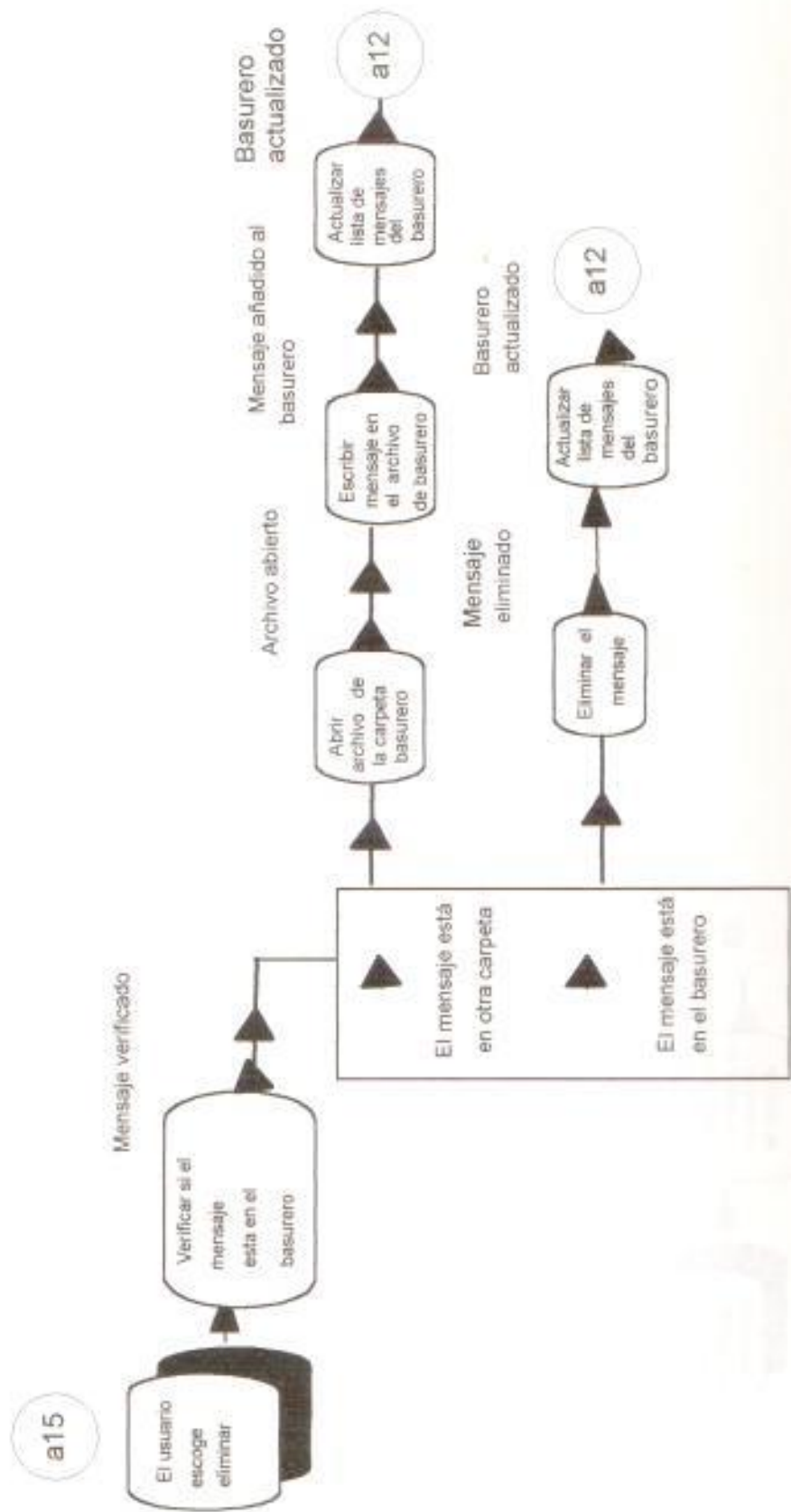


Fig. 22 Diagrama de Eventos . Sección 15

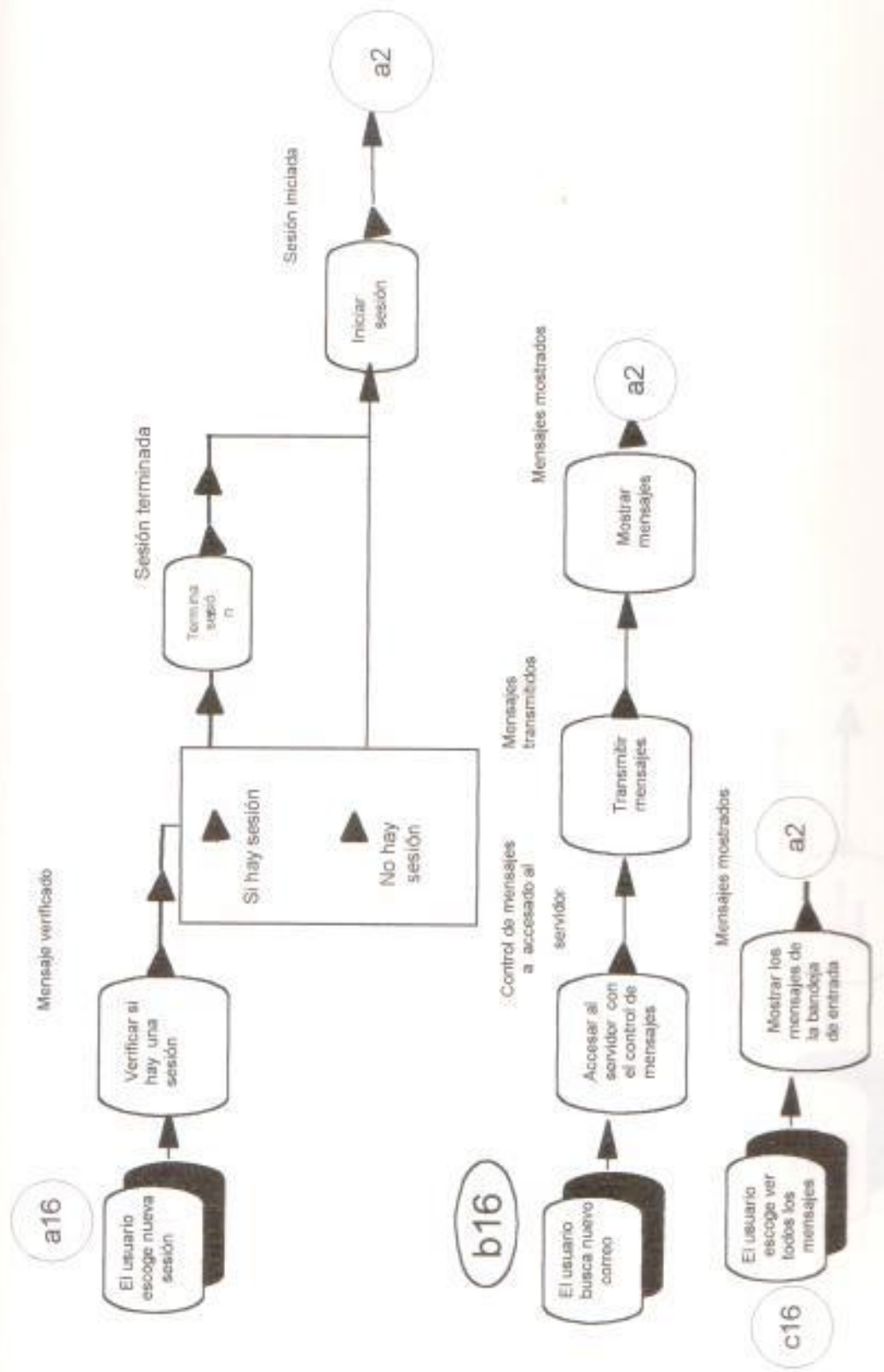


Fig. 23. Diagrama de eventos. Sección 16

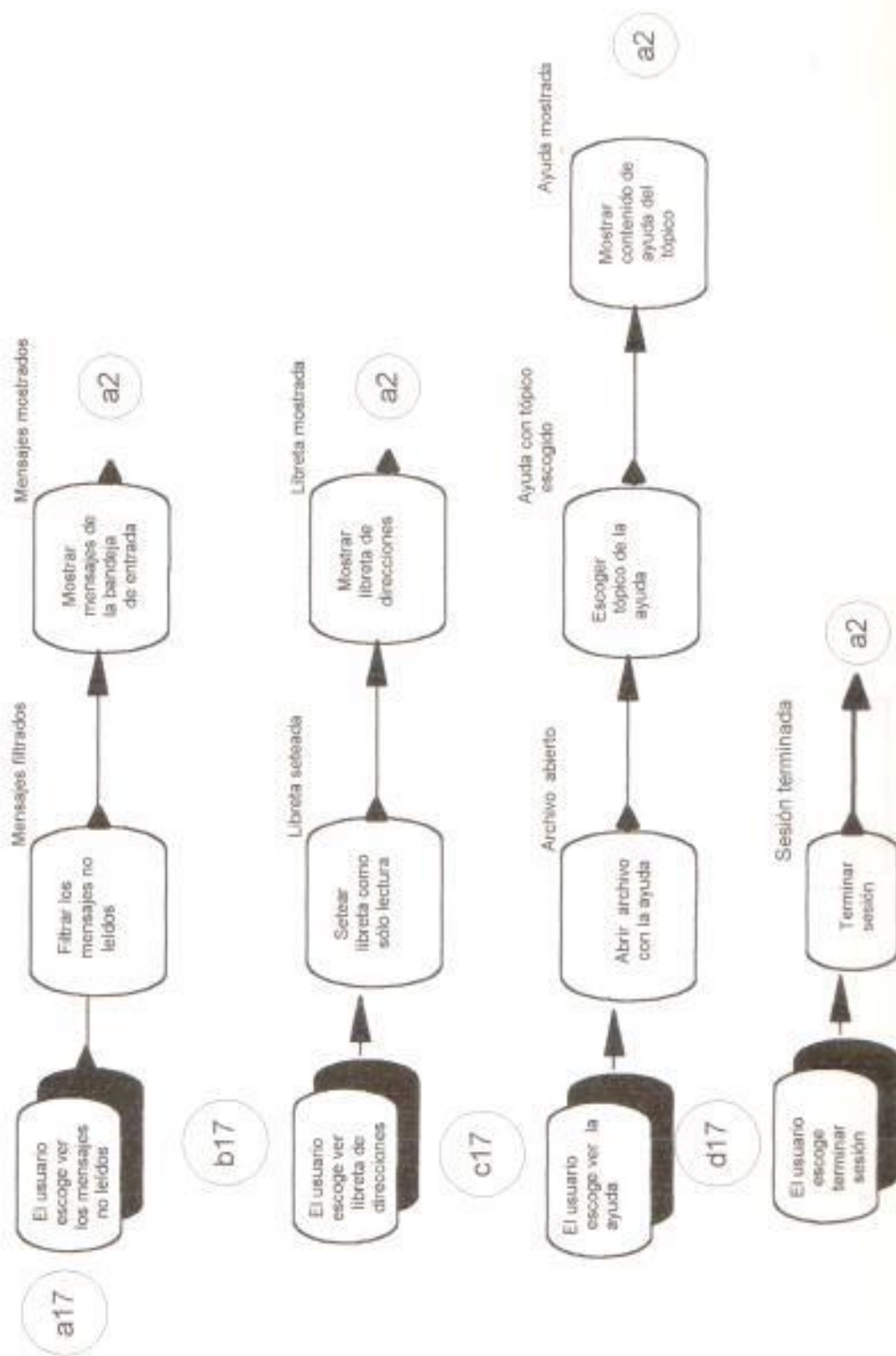


Fig. 24. Diagrama de Eventos . Sección 17

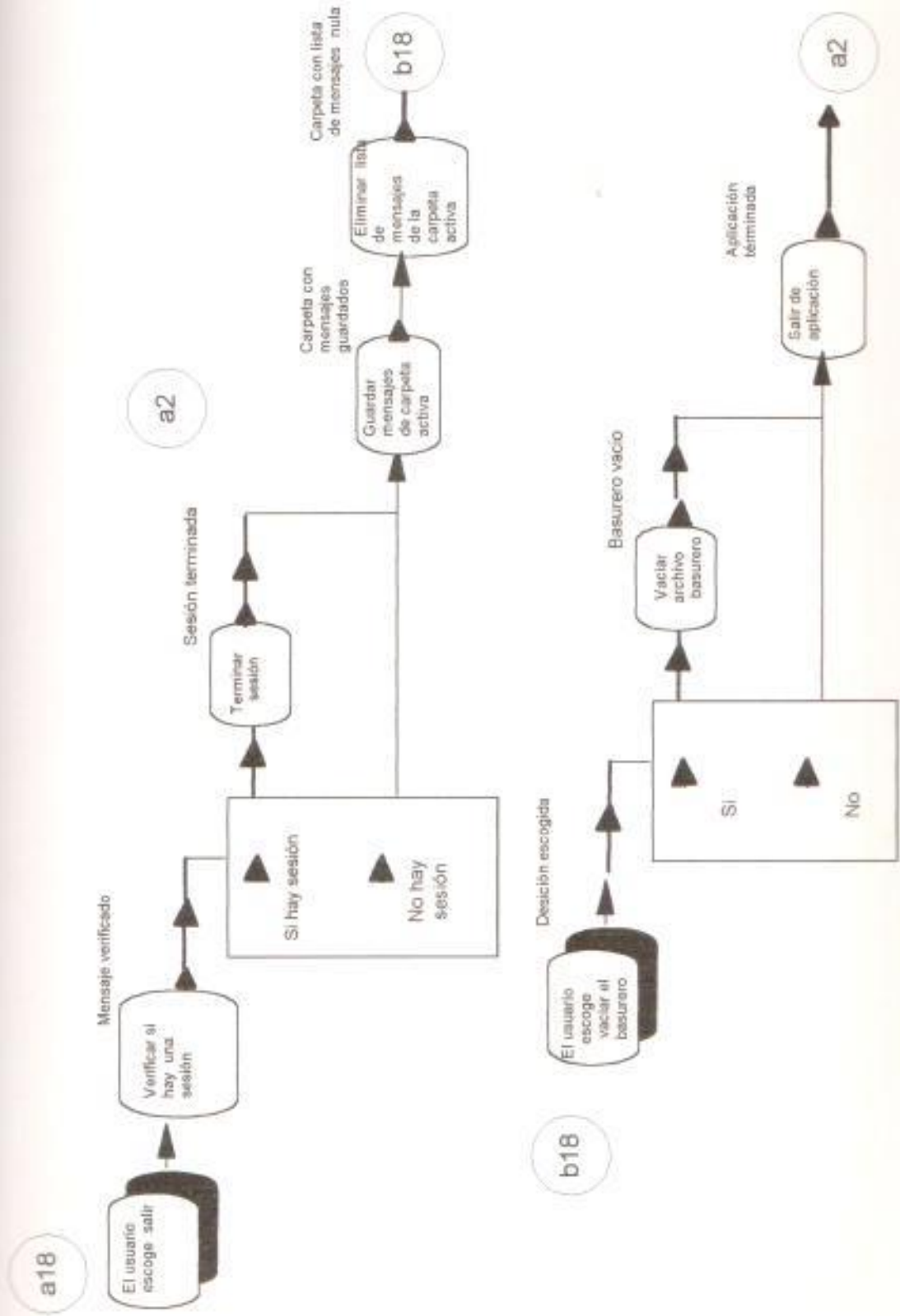


Fig. 25. Diagrama de Eventos . Sección 18

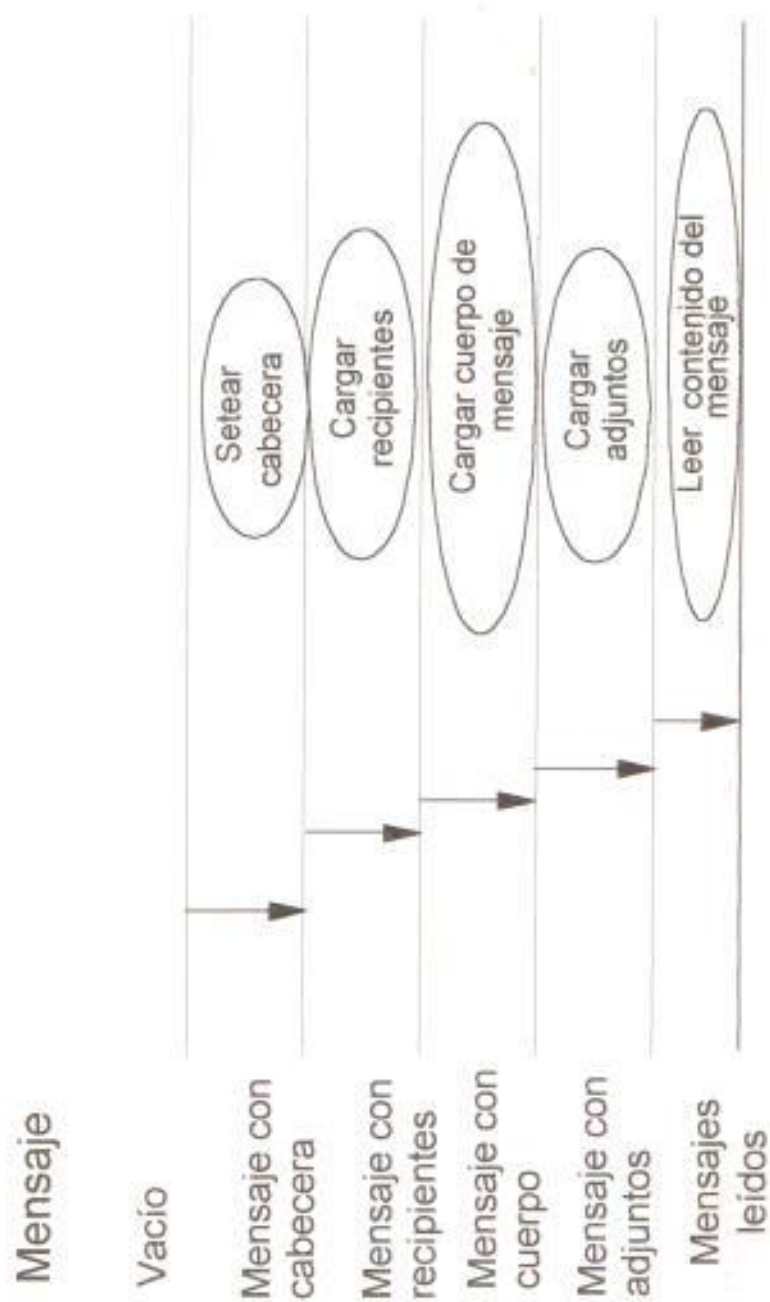


Fig.26 Diagrama de transición de estados . Objeto Mensaje.

Control de Sesión

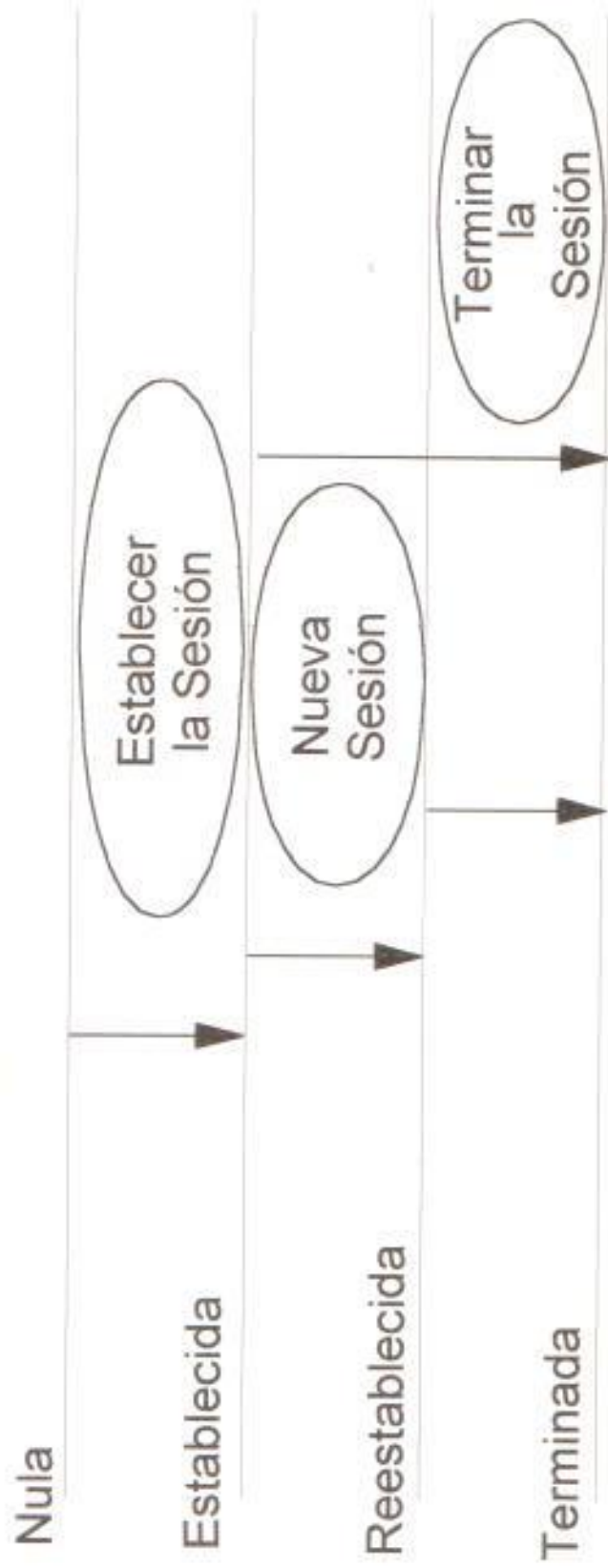


Fig. 27 Diagrama de Transición de Estados. Objeto Control de Sesión

Control de mensajes

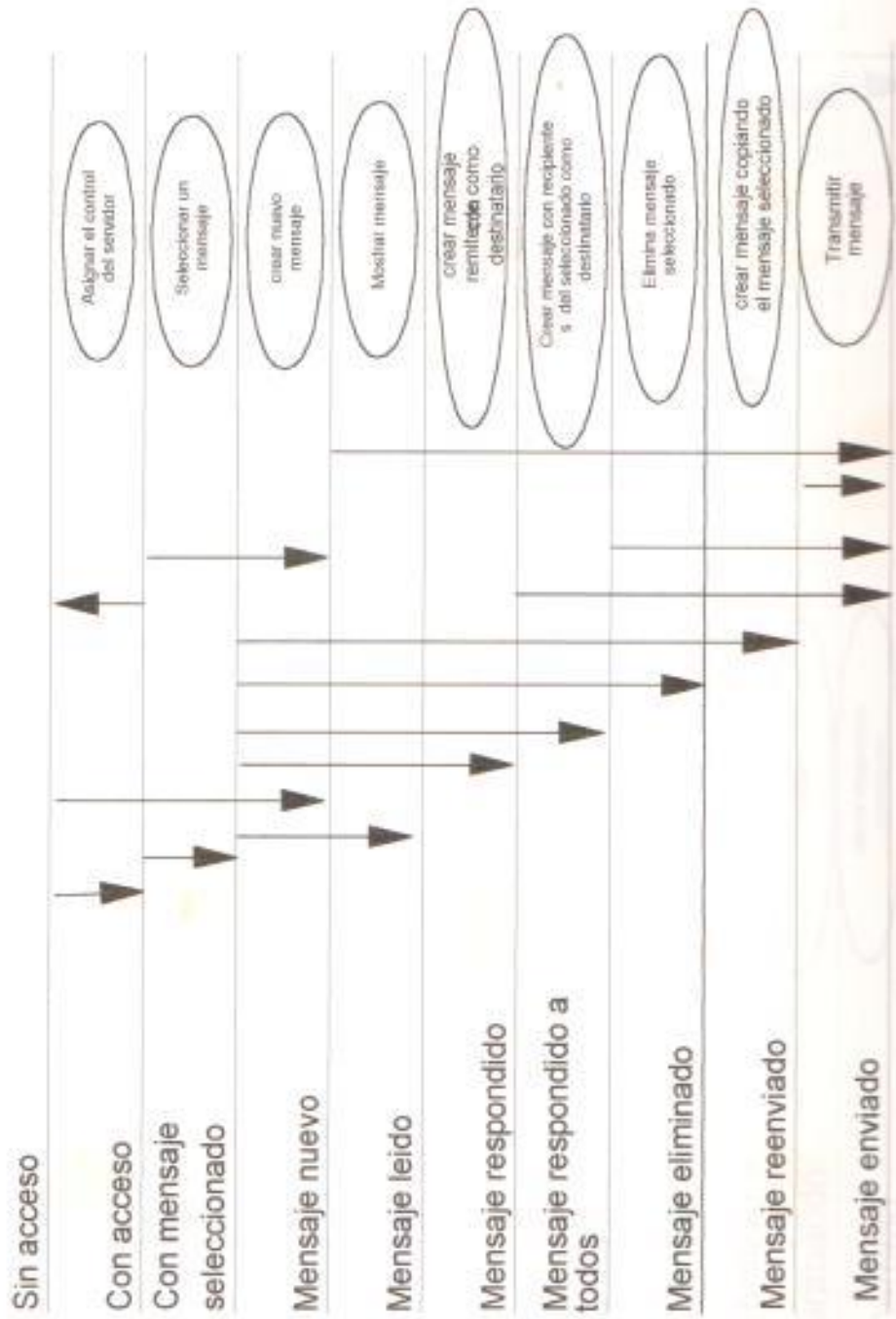


Fig. 28 Diagrama de Transición de Estados . Objeto Control de Mensaje

Reproductor

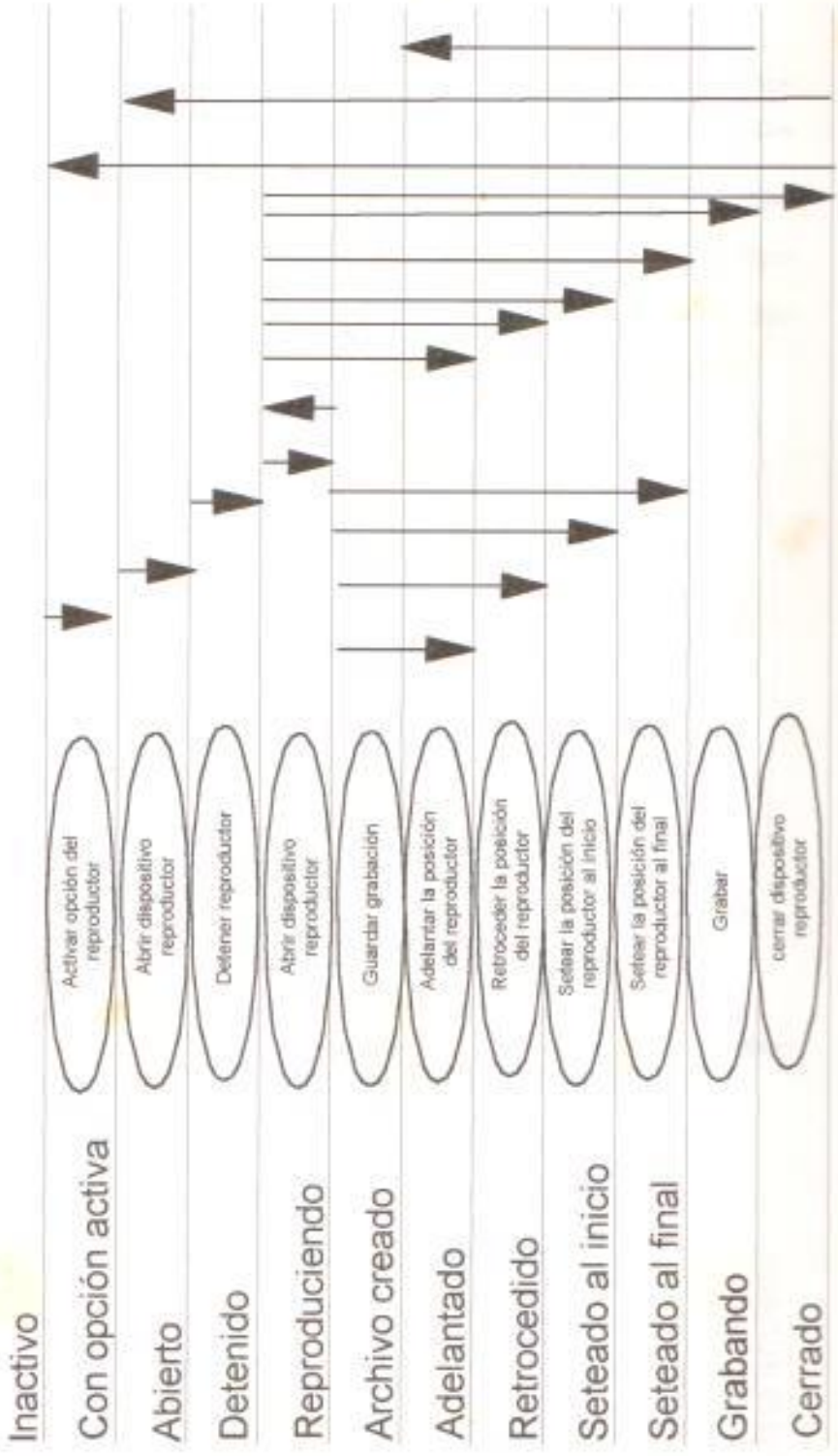


Fig. 29 Diagrama de Transición de Estados . Objeto Reproductor

Carpeta

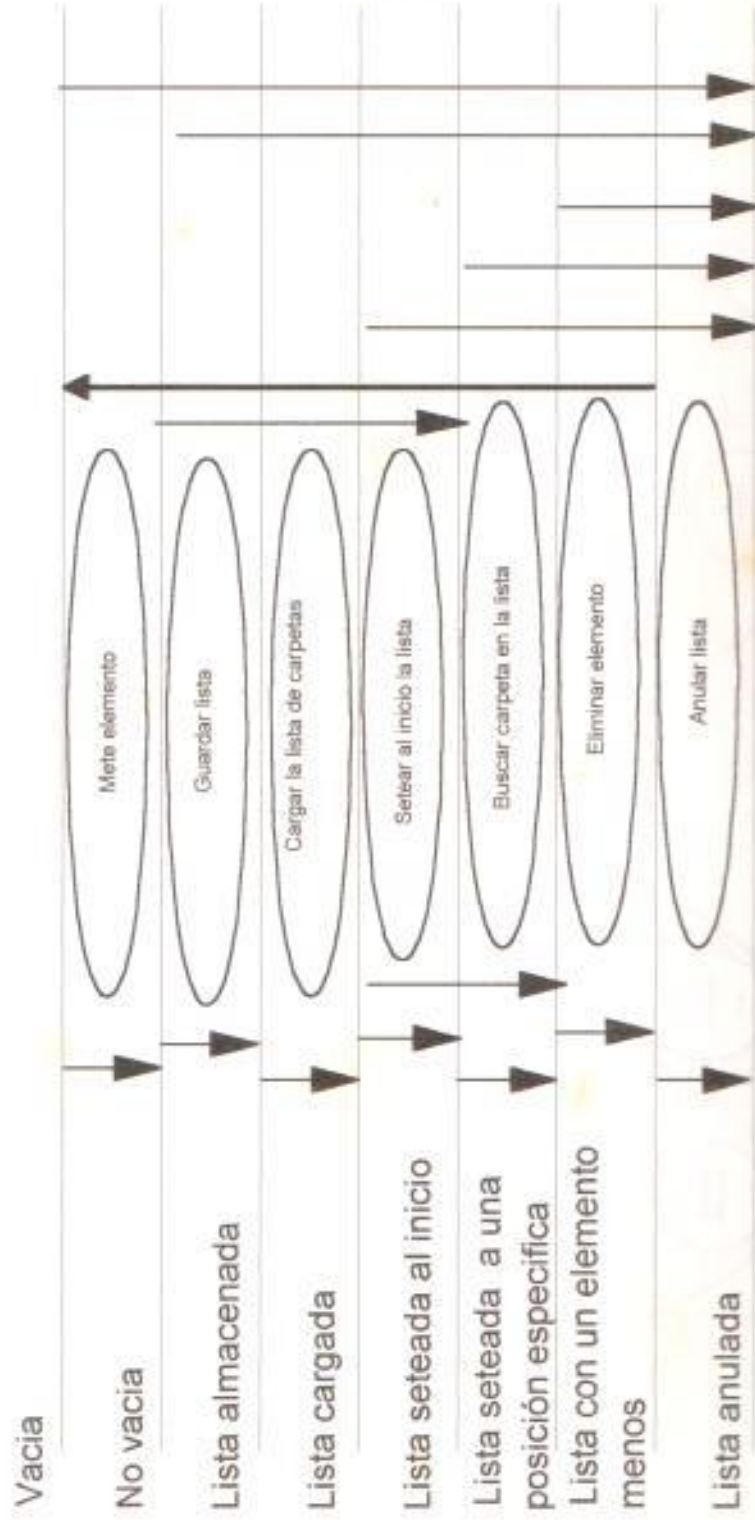


Fig. 30 Diagrama de Transición de Estados. Objeto Carpeta

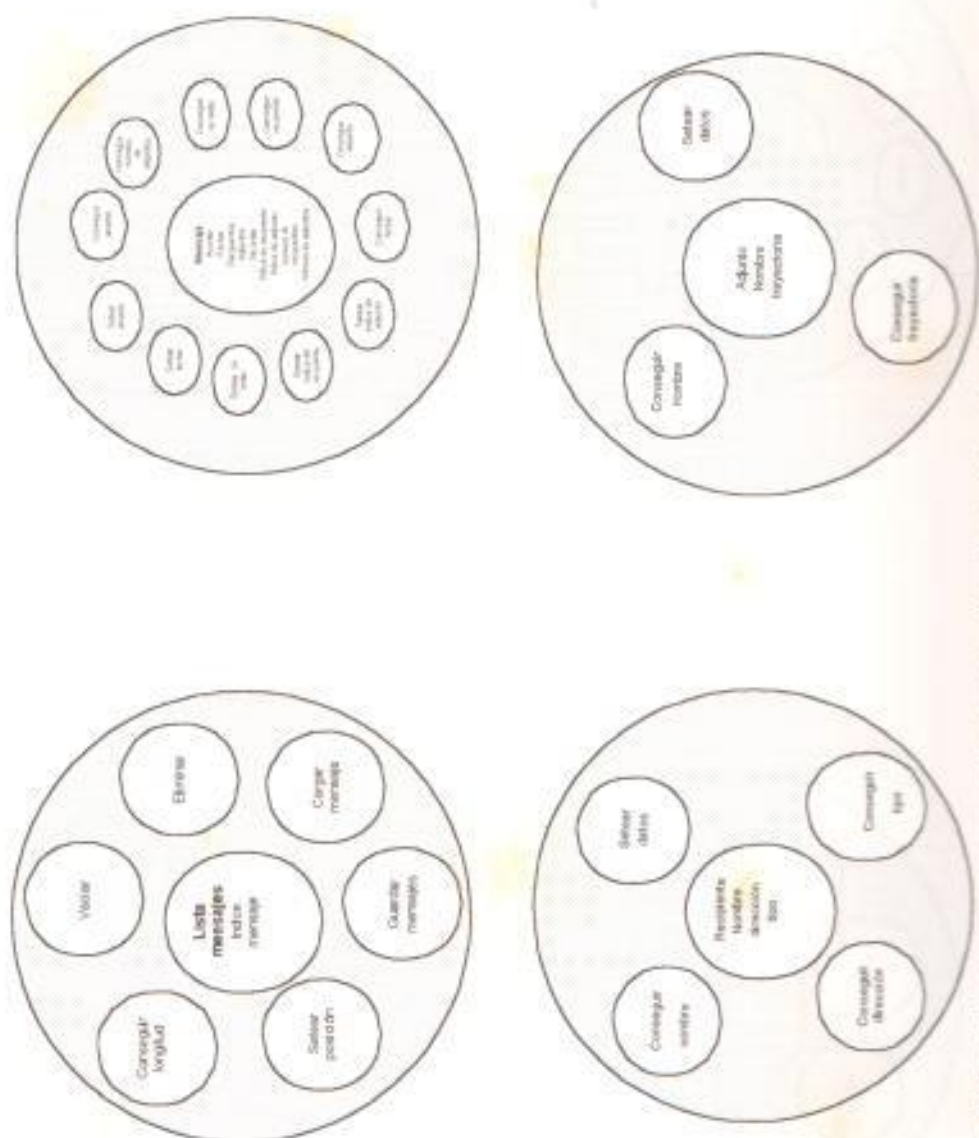


Fig. 31 Diseño de la estructura y comportamiento del objeto. Sección 1

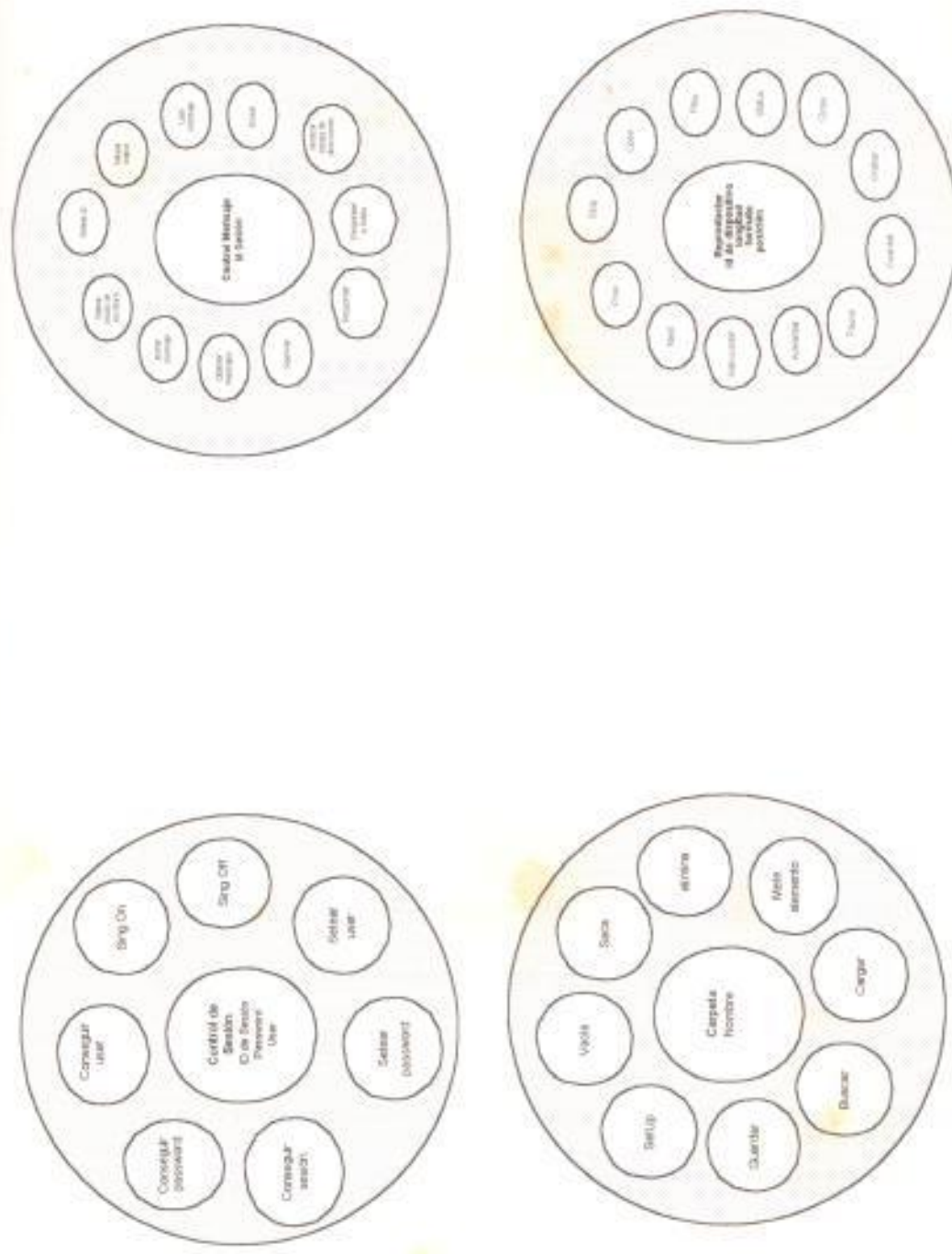


Fig. 32 Diseño de la estructura y comportamiento del objeto. Sección 2

Esta Clase forma parte de la estructura de la Clase CMensaje, y nos permite tener almacenados los archivos que forman parte de un mensaje los cuales son denominados adjuntos.

```
class CAttachmentMsg:public CObject  
{  
private:  
    CString m_NamePathFile;  
    CString m_NameFile;  
public:  
    CAttachmentMsg(CString path,CString name);  
    CString NameAttachment();  
    CString PathFile();  
};
```

La Clase Carpeta sirve para almacenar cada uno de los nombres de las carpetas que se crea el usuario para organizar los mensajes que desea guardar. Inicialmente se tienen dos carpetas definidas que son la Bandeja de entrada y el basurero (que es donde se guardan los mensajes que se eliminan temporalmente).

```

class CCarpeta
{
private :
    LisCar *Cabeza;
    LisCar *Posicion;//Posicion para sacar
    LisCar *Cola;
public:
    CCarpeta();
    CString SacaElemento();
    void MeteElemento(CString Elem);
    BOOL EliminaElemento(CString Elem);
    void Anula();
    BOOL Vacia();
    void SetUp();
    void Guardar(char * archi);
    void Cargar(char * archi);
    BOOL Buscar(CString archi);
};

```

La clase ClisMensaje fue elaborada con el objetivo de almacenar cada uno de los mensajes traídos desde la bandeja de entrada, para de esta manera manipular los mensajes.

class CLisMensaje

```
{  
private:  
    CObList Mensajes;  
    CMensaje *Temporal;  
    void receptor(CString recipiente);  
    void encabezado(CString cabeza);  
    void atachado(CString adjuntar);  
    void GuardaEncabezado(CString *mensaje);  
    void GuardaAdjuntos(CString *mensaje);  
    void GuardaRecipientes(CString *mensaje);  
public:  
    void CargarMensajes(char *archi);  
    void GuardarMensajes(char *archi);  
    BOOL EliminarMensaje(int i);  
    CMensaje *SetMsgIndex(int i);  
    int MensajesCont();  
    void AnulaLista();  
};
```

Con la Clase CmapiMessage(Control de Mensaje) se encarga de enviar y recibir mensajes a y desde el servidor, Para poder conseguir esto se debe estar conectado a el, y esto se da siempre

y cuando se haya obtenido el identificar que provee el Control de sesión que es quien consigue el enlace

```
class CMapiMessagesCtrl : public CWnd
```

```
{
```

```
protected:
```

```
    UINT ID de sesion
```

```
public:
```

```
    void Compose();
```

```
    void Copy();
```

```
    void Delete(const VARIANT& Obj);
```

```
    void Fetch();
```

```
    void Forward();
```

```
    void Reply();
```

```
    void ReplyAll();
```

```
    void ResolveName();
```

```
    void Save();
```

```
    void Show(const VARIANT& Details);
```

```
    void AboutBox();
```

```
    void Send(const VARIANT& Dialog);
```

```
};
```

La Clase CmapiMessage(Control de sesión).- Se encarga de conseguir el enlace entre la aplicación cliente y el servidor, para

esto se necesita especificar el usuario y contraseña con el fin de tener accesos a los mensaje almacenados en un buzón el cual pertenece a dicho usuario.

```
class CMapiSessionCtrl : public CWnd
```

```
{
```

```
protected:
```

```
public:
```

```
    CString GetUserName();  
    void SetUserName(LPCTSTR);  
    BOOL GetDownloadMail();  
    void SetDownloadMail(BOOL);  
    BOOL GetLogonUI();  
    void SetLogonUI(BOOL);  
    BOOL GetNewSession();  
    void SetNewSession(BOOL);  
    short GetAction();  
    void SetAction(short);  
    long GetSessionID();  
    void SetSessionID(long);  
    CString GetPassword();  
    void SetPassword(LPCTSTR);
```

Operaciones

public:

```
void SignOn();  
void SignOff();  
void AboutBox();
```

```
};
```

Esta clase permite almacenar toda la información que es parte de un mensaje, para de esta manera poder manipular con facilidad cada uno de los elementos que forma un mensaje, así como : Los recipientes, Los adjunto, el subject y el mismo cuerpo del mensaje.

class CMensaje:public CObject

```
{
```

private:

```
int m_NumRecip;  
int m_NumAdj;  
CString m_Date;  
BOOL m_UnReady;  
CString m_Subject;  
POSITION RecipIndex;  
POSITION AttachmentIndex;
```

public:

```
CObList Recipientes;  
CObList Adjuntos;  
CMensaje(int numr,int numa);  
int GetNumAttachment();  
int GetNumRecip();  
void SetDate(CString date);  
CString GetDate();  
void SetSubject(CString subject);  
CString GetSubject();  
void SetUnReady(BOOL unready);  
BOOL GetUnReady();  
void SetRecipIndex(POSITION index);  
POSITION GetRecipIndex();  
void SetAttachmentIndex(POSITION index);  
void InsertRecip(CString name,CString adress,short tipo);  
CRecipMsg *GetRecipMsg();  
void DeleteRecip(POSITION index);  
void InsertAttachment(CString path,CString name);  
CAttachmentMsgMsg *GetAttachment();  
void DeleteAttachment(POSITION index);  
CString GetOrigName();  
};
```

La clase forma parte de la estructura de CMensaje, y nos permite almacenar los nombres y las direcciones de los destinatarios de los remitentes.

```
class CRecipMsg:public CObject  
{  
  
private:  
    CString m_Name;  
    CString m_Adress;  
    int m_Tipo;  
  
public:  
    CRecipMsg(CString name,CString adress,short tipo);  
    CString NameRecip();  
    CString AdressRecip();  
    short TipeRecip();  
  
};
```

La clase Creproductor permite abrir un dispositivo multimedia con un archivo que tenga un formato de audio, para luego poder realizar cualquiera de las acciones especificadas por las operaciones, así como: Play, Stop, Retrocede, Adelantar, Grabar, etc.

```
class CReproductor
```

```
{
```

```
  private:
```

```
    UINT wDeviceID;  
    DWORD dwReturn;  
    BOOL PauRes;  
    DWORD Longitud;  
    DWORD Formato;  
    DWORD Posicion;
```

```
  public:
```

```
    CReproductor();  
    int Status;  
    void Play();  
    BOOL OpenFile(const char* file, const char* Dispositivo);  
    void Close();  
    void Stop();  
    void Prev();
```

```
void Next();  
BOOL Pause();  
void Rec();  
void Retroceder(DWORD Paso);  
void Adelantar(DWORD Paso);  
void Guardar(const char* file);  
UINT GetID();  
DWORD GetStatus(DWORD Flag);  
  
};
```



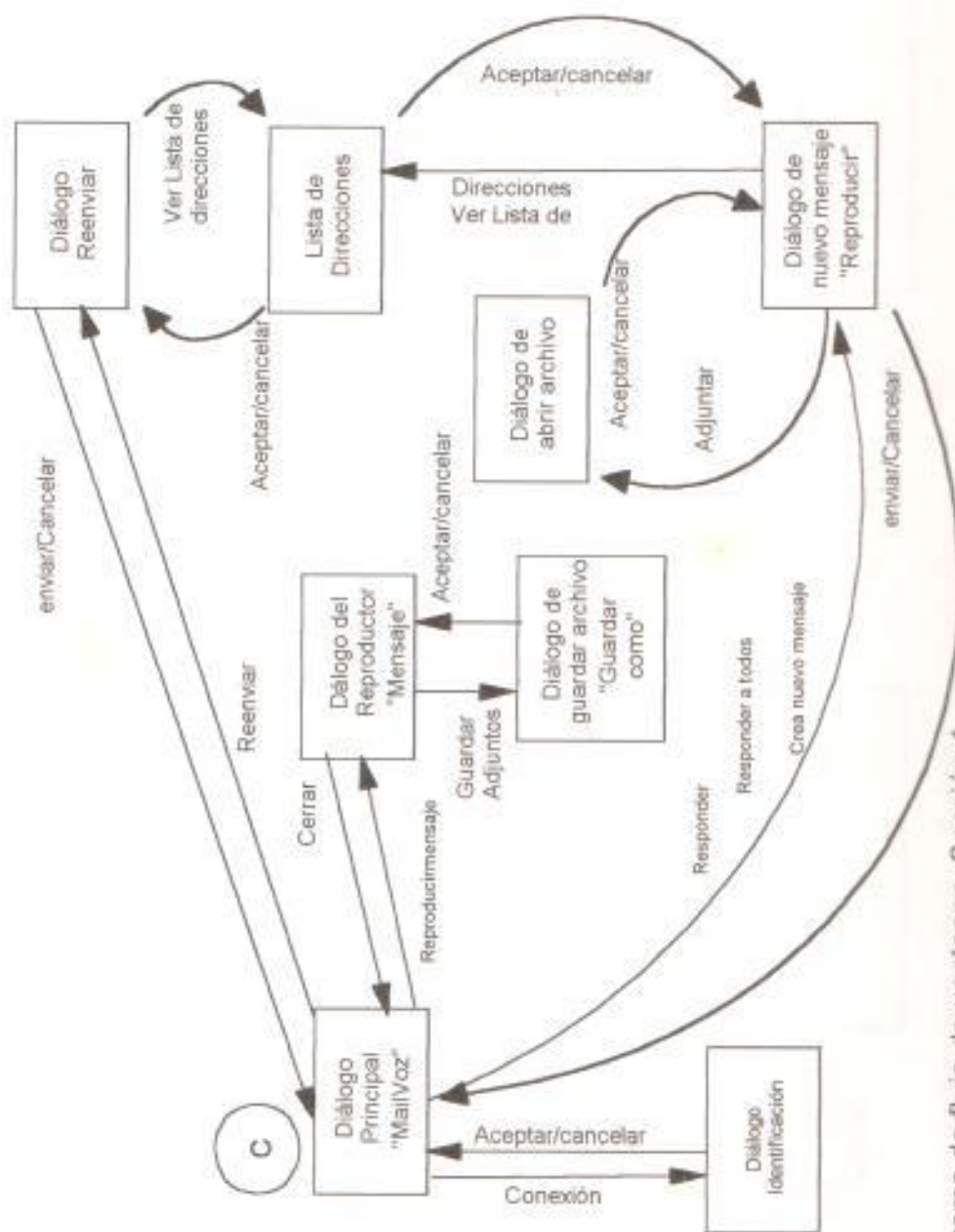


Fig 33. Diagrama de flujo de ventanas. Sección 1

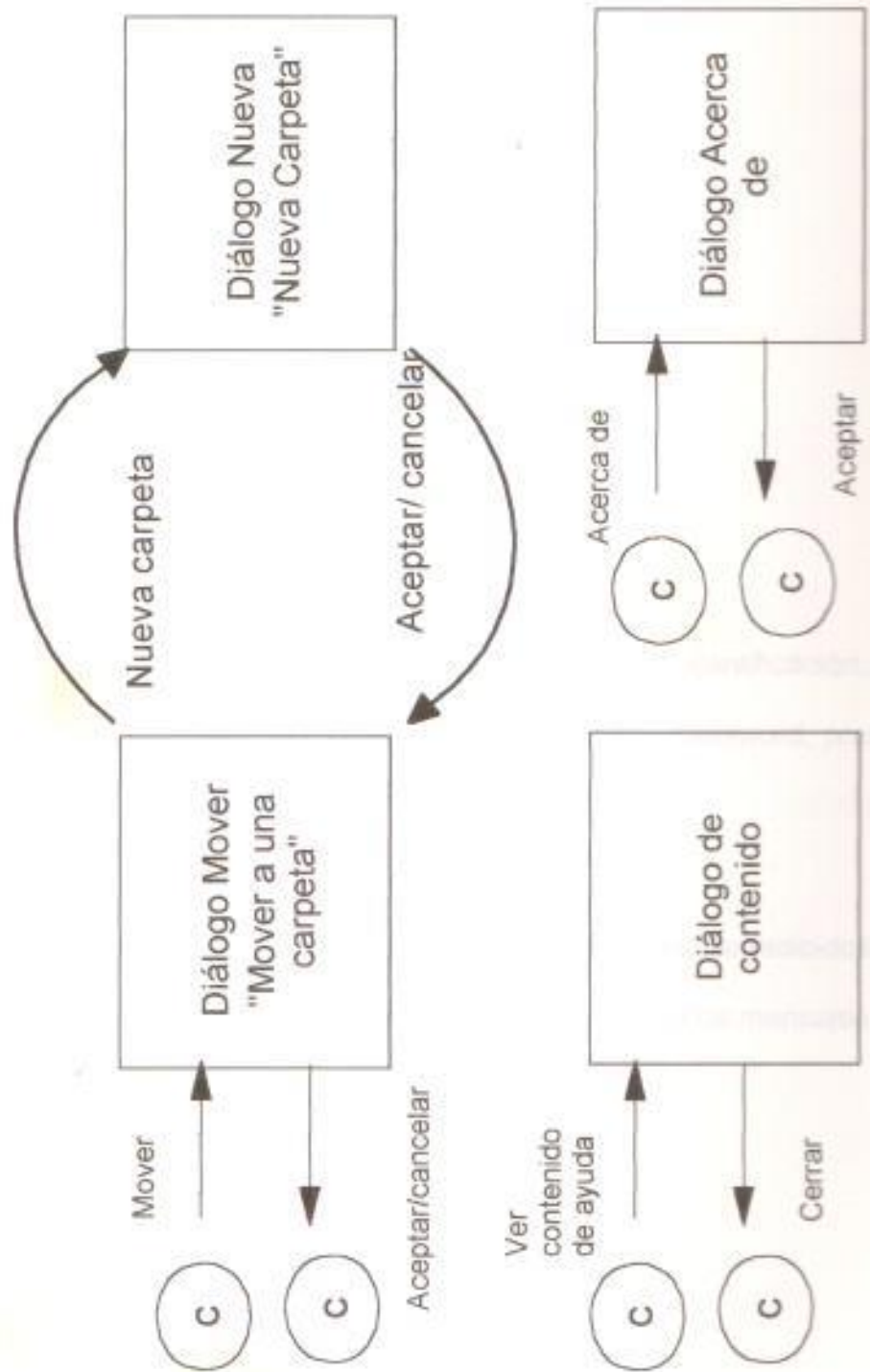


Fig 34. Diagrama de flujo de ventanas . Sección 2

Nombre del Grupo de Prueba: Conexión.

- Número de prueba: 1
- Pre-requisitos :
 - El programa recién se inicia
 - No hay ninguna sesión establecida.
- Instrucciones de configuración:
 - El usuario tiene cuenta en el servidor.
 - Tiene 7 mensajes en su buzón.
- Instrucciones de la Prueba:
 - El usuario ejecuta el comando *login*.
 - El sistema responde con el diálogo de identificación.
 - El usuario ingresa su identificación y password, presiona aceptar.
- Comportamiento Aceptable:
 - La lista de mensajes muestra los mensajes recibidos.
 - La barra de estado muestra la cantidad de mensajes.

☞ Número de prueba: 2

- Pre-requisitos:

Hay una sesión establecida

- Instrucciones de configuración:

La lista muestra 3 mensajes de la bandeja de entrada.

- Instrucciones de Prueba:

El usuario ejecuta el comando *nueva sesión*.

El sistema responde mostrando el dialogo de identificación.

El usuario ingresa su identificación y password, presiona aceptar.

- Comportamiento Aceptable:

La lista de mensajes se actualiza con los mensajes recibidos.

La barra de estado muestra la cantidad de mensajes.

☞ Nombre del Grupo de Prueba: Reproducción de mensajes.

- Número de prueba: 3

- Pre-requisitos :

Se ha seleccionado un mensaje.

- Instrucciones de configuración:

El cuerpo del Mensaje es de formato de audio.

Hay 2 recipientes Con copia.

El mensaje contiene 3 adjuntos

- Instrucciones de la Prueba:

El usuario ejecuta el comando *reproducir*.

El sistema responde con el diálogo de Reproducir mensaje.

El usuario reproduce el mensaje, retrocede, avanza, hace pausa, reproduce, stop.

El usuario selecciona un adjunto y presiona guardar.

El sistema responde con el diálogo de guardar archivo.

El usuario selecciona el directorio, le asigna un nombre y acepta.

El sistema responde cerrando el diálogo.

El usuario cierra el diálogo reproducción.

- Comportamiento Aceptable:

El dialogo muestra el nombre del remitente, los 2 Con copia, el asunto y la lista de adjuntos indicando que existen 3.

Al reproducir se escuchó el mensaje y respondió a todas las acciones.

El adjunto seleccionado se encuentra en el directorio especificado con el nombre asignado.

Al cerrar el diálogo el mensaje se encuentra en la lista con estatus leído.

☞ Número de prueba: 4

- Pre-requisitos:

Se ha seleccionado un mensaje.

- Instrucciones de configuración:

El cuerpo del Mensaje *no* es de formato de *audio*.

Hay 5 recipientes Con copia.

El mensaje contiene 1 adjunto.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *reproducir*.

El sistema responde con una advertencia sobre el formato del mensaje.

El usuario presiona aceptar.

El sistema responde con el diálogo de Reproducir mensaje.

El usuario selecciona un adjunto y presiona guardar.

El sistema responde con el diálogo de guardar archivo.

El usuario selecciona el directorio, le asigna un nombre y acepta.

El sistema responde cerrando el diálogo.

El usuario cierra el diálogo reproducción.

- Comportamiento Aceptable:

El dialogo muestra el nombre del remitente, los 5 Con copia, el asunto y la lista de adjuntos indicando que existe 1.

No se pudo reproducir.

El adjunto seleccionado se encuentra en el directorio especificado con el nombre asignado.

Al cerrar el diálogo el mensaje se encuentra en la lista con estatus leído.

- ☞ Nombre del Grupo de Prueba: Crear nuevo mensaje.

- Número de prueba: 5

- Pre-requisitos :

Se ha establecido una sesión.

- Instrucciones de configuración:

Se intentará enviar a 1 Destinatario original que no existe.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *nuevo mensaje*.

El sistema responde con el diálogo de Nuevo mensaje.

El usuario escribe 2 recipientes Destinatarios originales.

El usuario presiona *comprobar*.

El sistema responde advirtiéndolo de un error en los recipientes.

El usuario utiliza la libreta de direcciones para agregar 3 Con Copia y un Destinatario original.

El usuario graba el mensaje, stop, retrocede y vuelve a grabar.

El usuario detiene la grabación, la retrocede y la reproduce.

El usuario presiona *adjuntar*.

El sistema responde con el diálogo de Abrir archivo.

El usuario selecciona el directorio, el nombre y acepta.

El sistema responde cerrando el diálogo.

El usuario presiona *enviar* y el diálogo nuevo mensaje se cierra.

- Comportamiento Aceptable:

El dialogo nuevo al iniciarse sólo permite grabar.

Al reproducir se escuchó el mensaje y respondió a todas las acciones.

La libreta de direcciones mostraba las direcciones del servidor y se podía utilizar los comandos *con copia y para*.

El nombre del archivo adjuntado se muestra en la lista de adjuntos y se indica la cantidad.

Al cerrar el diálogo el mensaje se guardó en la bandeja de salida.

No hubo error en el envío.

• Nombre del Grupo de Prueba: Responder un mensaje.

- Número de prueba: 6

- Pre-requisitos :

Se ha establecido una sesión.

Se ha escogido un mensaje.

- Instrucciones de configuración:

Enviar con copia a 2 nuevos recipientes válidos.

Se enviará 3 adjuntos de tamaños mayores a 1Mb

- Instrucciones de la Prueba:

El usuario ejecuta el comando *responder mensaje*.

El sistema responde mostrando el diálogo nuevo mensaje, con el recipiente *Para* y el texto de *Asunto*, seteados con los del mensaje escogido. Además el texto del asunto empieza con la palabra *Re:* .

El usuario escribe 2 recipientes Con copia.

El usuario presiona *comprobar*.

El sistema pasa inadvertido al comando *comprobar* pues los recipientes son válidos.

El usuario graba el mensaje, stop, retrocede y vuelve a grabar.

El usuario presiona *adjuntar*.

El sistema responde con el diálogo de Abrir archivo.

El usuario selecciona el directorio, el nombre buscando archivos mayores a 1Mb y acepta.

El sistema responde cerrando el diálogo.

El usuario vuelve a adjuntar 2 archivos más con las mismas características.

El sistema responde de la forma ya descrita la primera vez.

El usuario presiona *enviar* y el diálogo nuevo mensaje se cierra.

- Comportamiento Aceptable:

El dialogo nuevo al iniciarse sólo permite grabar.

Al reproducir se escuchó el mensaje y respondió a todas las acciones.

El nombre de los archivos adjuntados se muestra en la lista de adjuntos uno a continuación de otro y se indica la cantidad de estos que en este caso será 3.

Al cerrar el diálogo el mensaje se guardó en la bandeja de salida.

No hubo error en el envío.

• Nombre del Grupo de Prueba: Responder un mensaje a todos.

- Número de prueba : 7

- Pre-requisitos :

Se ha establecido una sesión.

Se ha escogido un mensaje.

- Instrucciones de configuración:

El mensaje escogido para responder tiene 2 destinatarios Con Copia.

No se enviará ningún adjunto.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *responder a todos*.

El sistema responde mostrando el diálogo nuevo mensaje, con el recipiente *Para* los recipientes *Con Copia* y el texto de *Asunto*, seteados con los del mensaje escogido respectivamente. Además el texto del asunto empieza con la palabra *Re:* .

El usuario graba el mensaje, se posiciona al inicio, reproduce, se posiciona al final y vuelve a reproducir.

El usuario presiona *enviar* y el diálogo nuevo mensaje se cierra.

- Comportamiento Aceptable:

El dialogo nuevo al iniciarse sólo permite grabar.

Al reproducir se escuchó el mensaje la primera vez que se reprodujo desde el inicio pero, no la segunda por estar en la posición final del cuerpo del mensaje.

La lista aparece vacía y la cantidad que se indica será 0.

Al cerrar el diálogo el mensaje se guardó en la bandeja de salida.

El mismo mensaje se envió satisfactoriamente a todos los destinatarios.

☛ Nombre del Grupo de Prueba: Reenviar mensaje.

- Número de prueba: 8

- Pre-requisitos :

Se ha establecido una sesión.

Se ha escogido un mensaje.

- Instrucciones de configuración:

Se usará como destinatario Con copia la dirección del usuario actual.

El mensaje escogido tiene 4 adjuntos de distinto tamaño y tipo.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *reenviar*.

El sistema responde mostrando el diálogo nuevo mensaje, con el reproductor con la opción grabar desactivada y el texto de *Asunto*, seteado con los del mensaje escogido. Además el texto del asunto empieza con la palabra *Rw:* .

El usuario reproduce el mensaje, retrocede, lo detiene con una pausa y lo vuelve a reproducir.

El usuario intentará adjuntar presionando el botón respectivo, pero, el sistema no responderá a este evento.

El usuario presiona *enviar* y el diálogo nuevo mensaje se cierra.

- Comportamiento Aceptable:

El dialogo nuevo al iniciarse sólo permite reproducir.

El mensaje que se escuche al reproducir será el mismo del mensaje escogido y todas las operaciones con el reproductor responderán como se espera.

La lista aparece llena con los adjuntos del mensaje escogido y la cantidad que se indica será 4 en este caso.

Al cerrar el diálogo el mensaje se guardó en la bandeja de salida.

El mismo mensaje envió satisfactoriamente al destinatario original y una copia al propio usuario actual que era el destinatario con copia.

☛ Nombre del Grupo de Prueba: Buscar correo nuevo.

- Número de prueba : 9

- Pre-requisitos :

Se ha establecido una sesión.

- Instrucciones de configuración:

Se ha acaba de enviar un nuevo mensaje con copia para el destinatario actual.

Y la lista de mensajes tiene 5 mensajes actualmente todos ya leídos.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *buscar nuevo correo*.

- Comportamiento Aceptable:

La lista de mensajes se actualiza con los mensajes recibidos y marca como no leído el nuevo mensaje recibido.

La barra de estado muestra la cantidad de mensajes en esta ocasión dice 6.

☛ Nombre del Grupo de Prueba: Ver mensajes no leídos.

- Número de prueba: 10

- Pre-requisitos :

La lista de mensajes está cargada.

- Instrucciones de configuración:

Se tienen los 6 mensajes de la prueba anterior donde hay uno no leído.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *ver no leídos*.

- Comportamiento Aceptable:

La lista de mensajes se actualiza con el único mensaje no leído.

La barra de estado muestra la cantidad de mensajes en esta ocasión dice 1.

☛ Nombre del Grupo de Prueba: Ver todos.

- Número de prueba: 11

- Pre-requisitos :

Se realizó la prueba anterior.

Y ningún otro cambio.

- Instrucciones de configuración:

Se tiene 1 mensaje no leído de la prueba anterior.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *ver todos*.

- Comportamiento Aceptable:

La lista de mensajes se actualiza con todos los mensajes leídos y no leídos.

La barra de estado muestra la cantidad de mensajes en esta ocasión dice 6.

• Nombre del Grupo de Prueba: Mover.

- Número de prueba: 12

- Pre-requisitos :

Tener la lista de mensajes cargada.

Se ha escogido un mensaje.

- Instrucciones de configuración:

Lista cargada con 5 mensajes.

Mueve a una nueva carpeta.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *mover*.

El sistema responde abriendo el dialogo de mover.

El usuario ejecuta el comando *nueva*.

El sistema responde abriendo el diálogo carpeta nueva.

El usuario ingresa el nombre de una carpeta existente y presiona aceptar.

El sistema cierra el diálogo carpeta nueva y no crea ninguna carpeta nueva pues ésta ya existe.

El usuario ejecuta nuevamente el comando *nueva* y esta vez escoge un nombre de carpeta que no exista.

El sistema responde creando la nueva carpeta con el nombre ingresado y deshabilita el botón *nueva*.

El usuario presiona aceptar.

El sistema responde moviendo el mensaje a la nueva carpeta y cerrando el diálogo de mover.

- Comportamiento Aceptable:

La lista de mensajes se actualiza con todos los mensajes menos el mensaje que acaba de ser movida, el cual es removido de la carpeta actual y copiado a la carpeta nueva.

La barra de estado muestra la cantidad de mensajes disminuida en uno, es decir, 4.

⇒ Número de prueba: 13

- Pre-requisitos:

Tener la lista de mensajes cargada.

Se ha escogido un mensaje.

Tener por lo menos 1 carpeta además de la actual para mover el mensaje.

- Instrucciones de configuración:

Lista cargada con 8 mensajes.

Mueve a una carpeta existente.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *mover*.

El sistema responde abriendo el dialogo de mover.

El usuario selecciona una carpeta y presiona OK.

El sistema responde moviendo el mensaje a la carpeta seleccionada y cerrando el diálogo de mover.

- Comportamiento Aceptable:

La lista de mensajes se actualiza con todos los mensajes menos el mensaje que acaba de ser movida, el cual es removido de la carpeta actual y copiado a la carpeta seleccionada.

La barra de estado muestra la cantidad de mensajes disminuida en uno, es decir, 7.

• Nombre del Grupo de Prueba: Eliminar.

- Número de prueba: 14

- Pre-requisitos :

Tener la lista de mensajes cargada.

Se ha escogido un mensaje.

- Instrucciones de configuración:

Lista cargada con 5 mensajes.

Carpeta actual no es el basurero.

El basurero está vacío.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *eliminar* por tres ocasiones.

- Comportamiento Aceptable:

La lista de mensajes se actualiza con todos los mensajes menos el mensaje que acaba de ser eliminado, el cual es removido de la carpeta actual y copiada al basurero, esto sucede cada vez que el usuario ejecutó *eliminar*.

La barra de estado muestra la cantidad de mensajes disminuida en uno, es decir, 2.

127 Nombre del Grupo de Prueba: Desconectarse.

- Número de prueba: 15

- Pre-requisitos :

Tener una sesión existente.

- Instrucciones de configuración:

La lista cargada es la bandeja de entrada.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *logout*.

- Comportamiento Aceptable:

La lista de mensajes se limpia.

La sesión se termina.

• Nombre del Grupo de Prueba: Salir.

- Número de prueba: 16

- Pre-requisitos :

Tener abierto el sistema.

- Instrucciones de configuración:

El basurero no está vacío.

Existe una sesión establecida.

- Instrucciones de la Prueba:

El usuario ejecuta el comando *salir*.

El sistema responde con un mensaje preguntando si elimina los mensajes del basurero.

El usuario presiona OK.

El sistema responde cerrando la aplicación.

- Comportamiento Aceptable:

La lista de mensajes se limpia.

El basurero, queda vacío.

La sesión se termina.

2.7.- PROBLEMAS DE IMPLEMENTACION Y ALGORITMOS UTILIZADOS

Esta sección describe los problemas de implementación que encontramos y los algoritmos en que fueron resueltos o las herramientas que se usaron.

- Cómo establecer la comunicación con el servidor.

Entre los requerimientos del sistema encontramos que debe servir para correo de Internet, correo en una red de área local, o incluso para la oficina de correo de Windows 95. Para esto necesitábamos elaborar varias interfaces de comunicación que se adapten al protocolo de comunicación utilizado por el servidor. La respuesta a nuestro problema fue utilizar una interface de comunicación provista por las funciones de mensajería API de Windows MAPI, las cuales utilizan los servicios de correo configurados en su sistema y establecen la comunicación con el servidor.

Investigando encontramos dos clases basadas en MAPI, la clase `MapiSessionCtrl` y `MapiMsgCtrl`. La primera provee una interface

de comunicación para identificarse en el servidor, ingresando su user y password, y realizar algunas otras tareas relacionadas, así como desconectarse. La segunda provee la interfaz de manejo de mensajes en el servidor, para enviar, leer, eliminar, etc., basándose en el Identificador de Sesión provisto por el primero. El Algoritmo de conexión es simplemente:

```
if sesion.GetIdSesion es NULL then
    sesion.SignOn( ); //Identificarse en el servidor
else
    sesion.SignOff( ); //Terminar la sesión
    sesion.SignOn( ); //Identificarse en el servidor
endif

CtrlMsg.SetSessionID(Sesion.GetSesionID); //Asignar al Control
de Msg el ID_Sesion
```

De allí en adelante, se trata de usar las facilidades de `MapiMsgCtrl` para manipular los mensajes.

- Cómo representar el cuerpo de los mensajes de voz.

El estándar del contenido de un mensaje, incluye un cuerpo de mensaje tipo texto, pero nuestro mensaje era de voz, por lo cual, no podíamos utilizarlo. Pero además del cuerpo del mensaje

tenemos archivos adjuntos, los cuales son contenidos en una lista de adjuntos.

Nuestro problema se resolvía asignando la primera posición en la lista de adjuntos al archivo de audio que contiene el mensaje de voz y considerando esto en los algoritmos de crear y reproducir el mensaje.

De esta manera un mensaje enviado por nuestro sistema, sería leído en cualquier otro correo que soporte adjuntos, y se podría escuchar abriendo el respectivo archivo.

El algoritmo para crear un mensaje debe impedir que un adjunto ocupe la primera posición:

```
posicion = 0;
if se adjunta y no se ha grabado then
    solicitar la grabación del mensaje
else
    if existe cuerpo then
        posicion=posicion +1;
        guardar el adjunto en la posicion;
        Hay adjuntos= verdadero;
```

```

else
    guardar cuerpo en posición; // Posición es cero
    Existe cuerpo= verdadero;
endif
endif
if se envía y no se ha grabado
    solicitar la grabación del mensaje;
else
    if existe cuerpo then
        Enviar
    else
        guardar cuerpo en posición; // Posición es cero
        Enviar;
    endif
endif
endif

```

El algoritmo de Reproducción del mensaje es:

```

if adjunto posición cero es audio then
    Reproductor.Open(Nombre del Archivo);
    Posición=1;
else
    Mensaje de formato no audio;

```



```
        Posicion=0;  
    endif  
    Cargar recipientes;  
    Cargar adjuntos; // Desde posición
```

- Cómo grabar y reproducir el cuerpo del mensaje.

En primer lugar elegimos un formato de tipo WAV para el archivo de audio. Estos tienen un tamaño de 43 Kb/seg. con una frecuencia de 22.050 Hz y 8bits. Esto nos brinda una buena calidad de sonido, pero a cambio de mucho espacio físico. Ej.: Un mensaje de un minuto tendría un tamaño de 2.580 Kb. Esto no es problema en la transmisión, porque se usa algoritmos para comprimirlos. Pero poco a poco van ocupando espacio en disco de una manera desagradable si no hay el mantenimiento adecuado de los mensajes.

Las funciones utilizadas para elaborar la clase reproductor son tomadas de la interface de Control Multimedia (MCI), las cuales proveen manipulación de los mensajes a un nivel intermedio. La función clave para enviar mensajes al dispositivo reproductor es:

```
mciSendCommand( UINT ID_Device, DWORD Comando,  
                DWORD Flags, (DWORD)(LPVOID) * Comand_Parms)
```

donde,

ID_Device es el identificador del dispositivo,

Comand es el Comando a ejecutarse que puede ser
MCI_OPEN, MCI_PLAY,

MCI_RECORD, etc.

Flags son banderas de error aplicadas a comandos específicos

Comand_Parms es la estructura que contendrá los datos devueltos por la función y son específicas de acuerdo al comando.

Como interface con el usuario utilizamos el control OLE MCICtrl, que tiene botones personalizados de Play, Stop, Rec, Rew, FF, Prev, Next, Pause y Eject. Estos botones pueden habilitarse y deshabilitarse, así como hacerse visibles o no.

- Cómo guardar los adjuntos.

Los adjuntos pueden ser de cualquier tamaño y se necesita de un algoritmo para poder guardarlos en disco. Utilizamos la clase *Cfile* para manipular el archivo de la siguiente forma:

```
CFile Fuente; // Archivo fuente
```

```
CFile Destino; // Archivo destino
```

```
char bufer [1000]; // bufer temporal para leer y escribir.
```

```

int size; // longitud leída

// Conseguir nombre y trayectoria del archivo destino
(destinofile);

VERIFY(Fuente.Open(fuentefile, Cfile:modeRead)); //Se abre el
archivo en modo lectura

VERIFY(Destino.Open(destinofile, CFile:modeCreate|CFile:modeWr
ite);

size=Fuente.Read (bufer, sizeof (bufer));

while size<>0 do
{
    bufer[size]='\0';
    Destino.Write(bufer, size);
    size=Fuente.Read (bufer, sizeof (bufer));
}

Fuente.Close( );
Destino.Close ( );

```

- Cómo guardar la lista de carpetas.

La lista de carpetas se guarda usando nuevamente la clase Cfile, usando como nombre de archivo el nombre predeterminado Carpetas.txt que se encontrará en el directorio de la aplicación. La información que se va almacenar no es otra cosa que los

nombres de las carpetas personales creadas por el usuario con el fin de organizar sus mensajes y sólo son de tipo Cstring. Se puede añadir que los nombres de las carpetas se guardan en el mismo orden que tiene en la lista de carpetas en memoria el cual es en orden alfabético ascendente el cual se respeta cada vez que un nuevo nombre de carpeta es ingresado siguiendo el presente algoritmo:

```
LisCar *temp; //puntero para indicar la posición actual en la lista de
carpetas
LisCar *Ant; //puntero que indica la posición anterior a temp
BOOL Exito;
if(Cabeza==NULL)
{
    Cabeza=crea_elemento(LisCar); //Crea el primer elemento de
la lista
    Cabeza->dato=Elem;
    Cabeza->sig=NULL;
    Cola=Cabeza;
}
else
{
```

```
temp=Cabeza; //Busca la posición donde insertar el nuevo
             elemento.
Exito=FALSE;
do
{
    if(temp->dato>Elem)
    {
        if(temp==Cabeza)
        {
            Cabeza=new(LisCar);
            Cabeza->dato=Elem;
            Cabeza->sig=temp;
        }
        else
        {
            Ant->sig=new(LisCar);
            Ant->sig->dato=Elem;
            Ant->sig->sig=temp;
        }
        Exito=TRUE;
    }
    else
    {
```

```

        Ant=temp;
        temp=temp->sig;
        if(temp==NULL)
        {
            Exito=TRUE;
            Cola->sig=new(LisCar);
            Cola->sig->dato=Elem;
            Cola->sig->sig=NULL;
            Cola=Cola->sig;
        }
    }
}while(Exito==FALSE);
}
Posicion=Cabeza;

```

- Cómo guardar la lista de mensajes.

La verdad es que fue un verdadero problema tener que encontrar la forma de guardar la estructura de un mensaje completo en el mismo archivo (con el nombre de la carpeta donde esta) y recuperarlo totalmente. Para ello decidimos escoger el siguiente formato para poder reconocer cada una de las partes que componen el mensaje:

Formato de mensaje en el archivo de texto:

Asunto; Fecha; no leído; número de recipientes; número de adjuntos[nombre de recipiente1; dirección del recipiente 1;tipo del recipiente1|nombre de recipiente 2...]{ruta de adjunto1 ; nombre de adjunto 1|nombre de adjunto 2...}

Para guardar la lista de mensajes, tomamos mensaje por mensaje, le aplicamos el formato anterior y los concatenamos para formar un sólo texto y guardarlo de una vez en el archivo en forma secuencial. Usamos el algoritmo siguiente para guardar la lista:

```
CFile Destino; // Archivo destino
char bufer [10000]; // bufer temporal para leer y escribir.
int size; // longitud leida
VERIFY(Destino.Open(nombre de carpeta,CFile::modeWrite);
        Destino.Write(bufer,size);
While Lista de Mensajes no este vacia
{
    mensaje=Sacar_mensaje( ); //Saca mensaje de la lista
    Formar_encabezado(texto,mensaje); //Une asunto,fecha y no
    leído
    texto=texto+'[';
    Formar_recipientes(texto,mensaje); //Une los recipientes
```

```
texto=texto+']+'+'{' ;
Formar_adjuntos(texto,mensaje); //Une los adjuntos
texto=texto+'}';
}
bufer=texto;
Destino.Write(bufer,size);
Destino.Close( );
```

Para cargar nuevamente la lista procedimos en el orden inverso al anterior y este algoritmo forma parte del método Cargar (archivo) de la clase CLisMensaje.

Capítulo III

MANUAL DEL USUARIO

3.1.- .-INTRODUCCION

Es un Correo por Voz que permite a un usuario grabar mensajes en formato WAV usando el micrófono y enviarlos a otro usuario conociendo su dirección que estará almacenada en el Administrador de Correo, dirección que se obtendrá gracias a una lista de usuarios que se formara en el momento de ejecutarse la aplicación desde el cliente. De la misma forma como enviamos mensajes en formato WAV, también el sistema facilitara que se pueda leer mensajes que le pertenezcan al usuario que se encuentra usando la aplicación cliente en el momento o sea que no pueda leer mensajes que no son de su propiedad.

3.1.1.-CONVENCIONES DEL MANUAL.-

En este Manual se han tomado las siguientes convenciones para referimos a pantallas, menú, Campos, etc.

Si se tiene:	Convención:
Menú	Negrillas
Nombre de Pantallas	<i>Negrillas</i>
Botones	<u>Negrillas</u>
Campos	[Negrillas]
Título de Frame	<Negrillas>
Casilla de Verificación	{Negrillas}

3.1.2.-DESCRIPCION DEL AMBIENTE EN QUE SE VA HA TRABAJAR.-

El ambiente con el que se esta trabajando es muy amigable. Cuando se arranca Mailvoz en su computadora, nos aparecerá una ventana que será la pantalla principal que constará de las siguientes partes:

- Barra de Menú Principal.
- Barra de iconos.
- Lista de mensajes.
- Lista de Carpetas.

-Barra de Status.



... de actividades y contenidos/
... y con cursos/
... una lista de cursos de la

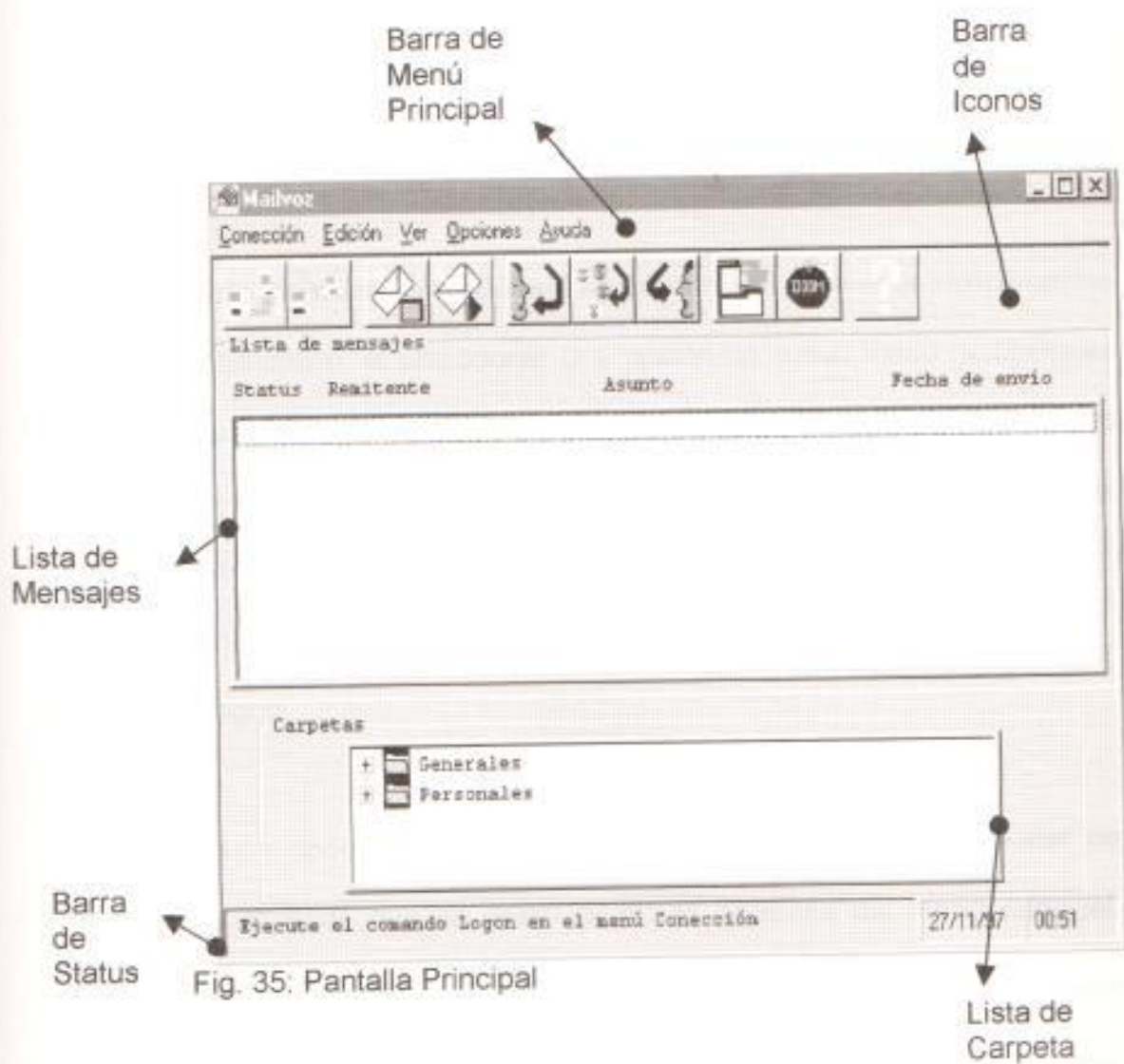


Fig. 35: Pantalla Principal

La **Barra de Menú Principal** consta de cinco opciones (órdenes) que aparecen en la parte superior de la ventana, y son: **Conexión**, **Edición**, **Ver**, **Opciones** y **Ayuda**. Para seleccionar una opción se lo puede hacer de dos maneras diferentes:

- Utilizar las teclas de movimiento de cursor → o ← para moverse por la barra de menú a la opción que se desea y pulsar la tecla Enter.
- Pulsar **Alt** más la tecla correspondiente a la primera letra de la opción: C, E, V, O, A. Por ejemplo Alt+C

Las opciones del menú que se utilizan son de menús desplegables (*pull-down*), al seleccionar una opción un nuevo menú de opciones aparece en la pantalla en una lista vertical, inmediatamente debajo de la orden del menú principal. Por ejemplo, al seleccionar **Conexión**, se desplegará el siguiente menú:

Logon

Logoff

Estas opciones del menú desplegable se pueden seleccionar de igual modo que se seleccionan las opciones del menú principal. Utilice las flechas ↑ o ↓, sitúese en la opción deseada y pulse Enter, o simplemente pulse Alt más la tecla en negrilla de la opción. Por ejemplo Alt+L.

Estando situado en una opción de un determinado menú se puede salir del mismo y retornar al menú jerárquico de nivel superior pulsando la tecla **Esc** (Escape), una vez por cada submenú.

En la **Barra de Iconos** tenemos iconos que ejecutan lo siguiente (Según el orden en el que están): **Conectarse**, **Desconectarse**, **Nuevo Correo**, **Reproducir correo**, **responder un mensaje seleccionado**, **Responder a todos**, **Reenviar**, **Copiar a una carpeta**, **Eliminar mensaje seleccionado** y **Ayuda**.

La **Lista de mensajes** se colocaran los mensajes que recibamos y también nos dará información de ellos como: **Status**, que nos indica si el mensaje es nuevo o ya a sido leído; **Remitente**, quien envió el mensaje; **Asunto**, de que se va ha tratar el mensaje, y por último **Fecha de envío**, en que fecha fue enviado el mensaje.

La **Lista de Carpetas** tendrá dos tipos de carpetas: **Generales**, son las que estarán en el Servidor y **Personales**, estarán en el Cliente.

Y por último se tiene a la **Barra de Status** que está dividida de tres partes, la primera parte nos indicará: **Que es lo que se tiene que hacer primero** o **el número de mensajes que se tiene**; y la

segunda parte nos indica la **fecha de la máquina** y la tercera parte la **hora de la máquina**.

3.1.3.- PREGUNTAS SOBRE MAILVOZ:

3.1.3.1.- ¿Qué contiene una Lista de Mensajes?

En esta lista nos aparecen todos los mensajes que nos han enviado, al seleccionar un mensaje en esta lista lo podemos reproducir, borrar o moverlo a una carpeta.

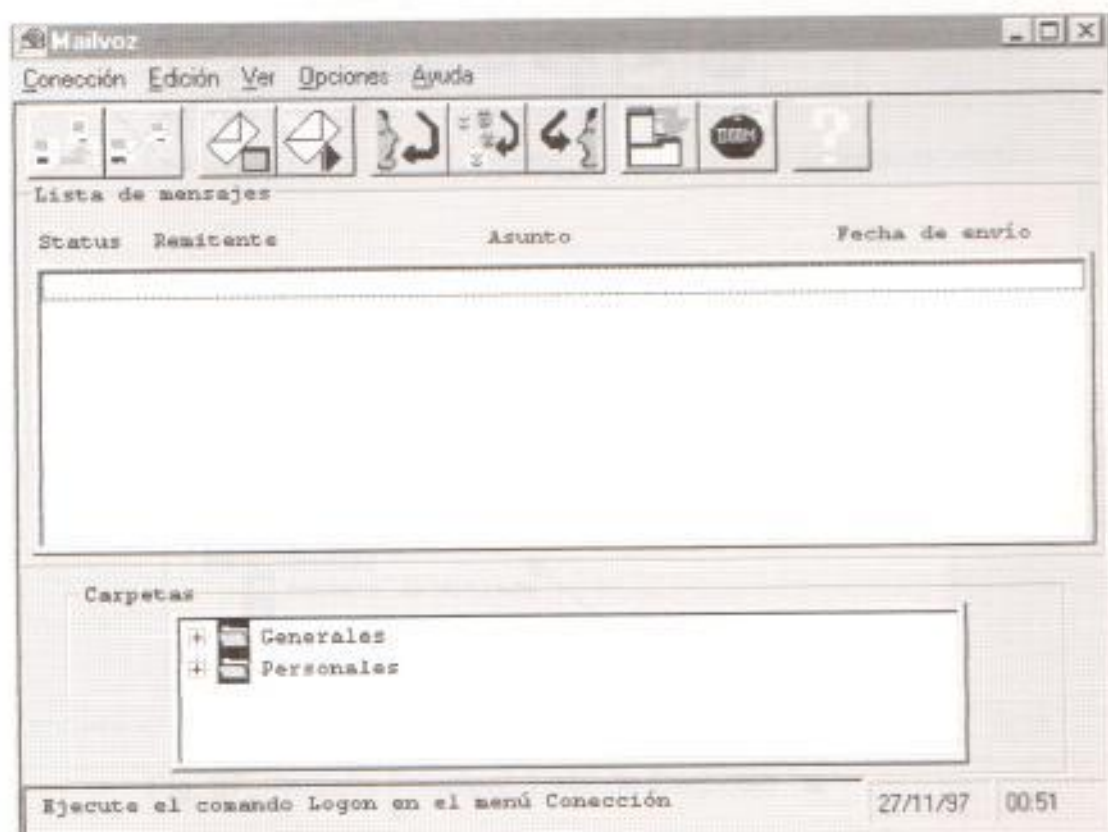


Fig. 36: Lista de mensajes

En la <Lista de mensajes> se puede obtener la siguiente información del mensaje:

Status: Indica si el mensaje es nuevo (→) o ya ha sido leído.

Remitente: Nombre del usuario que nos ha enviado el mensaje.

Asunto: De que se trata el mensaje.

Fecha de envío: Fecha en la cual se envió el correo.

3.1.3.1.- ¿Qué contienen las Carpetas?

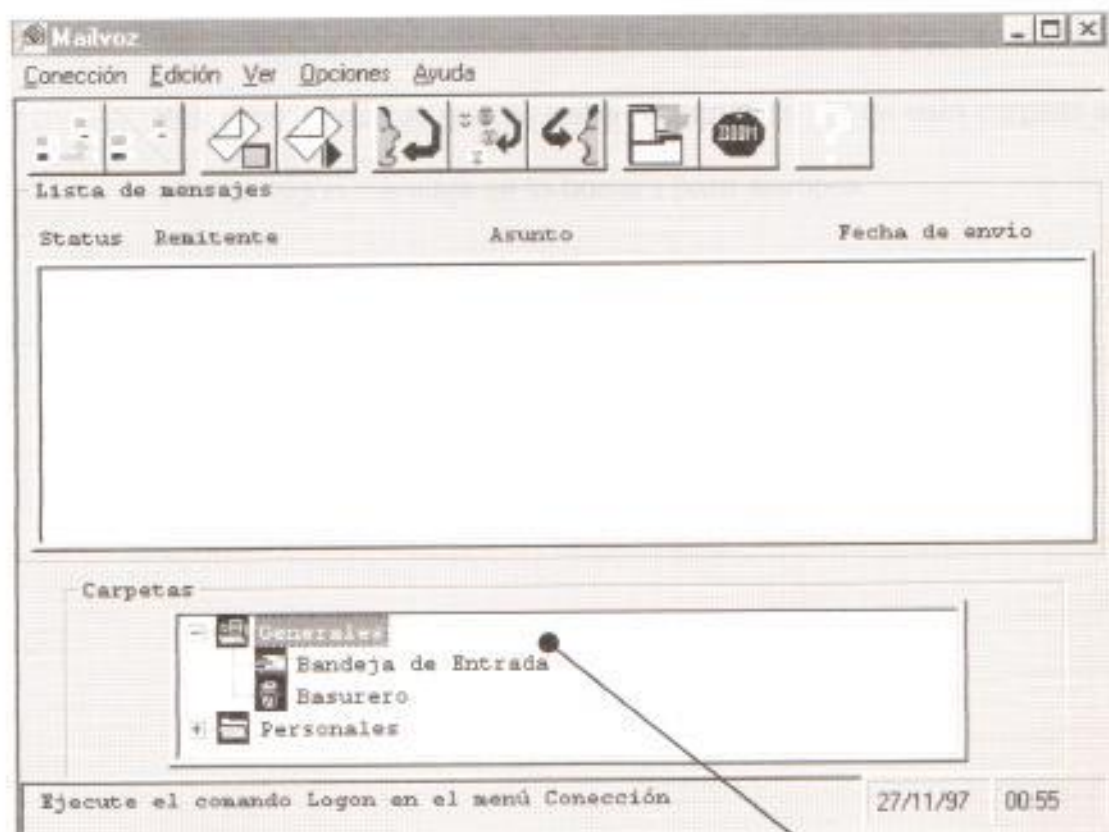


Fig. 37: Pantalla Principal, Lista de Carpetas

Lista de Carpetas
Generales

En la parte de <Carpetas> se puede observar dos tipos de carpetas:

- **Generales:**

Esta constituido de dos carpetas:

- Bandeja de Entrada

Esta carpeta se encuentra en el Servidor y se colocan los nuevos mensajes.

- Basurero

Esta carpeta se encuentra en la máquina Cliente y aquí se colocan todos los mensajes que se hallan eliminados. Si de esta carpeta se elimina algún mensaje se lo borrara para siempre.

- Personales

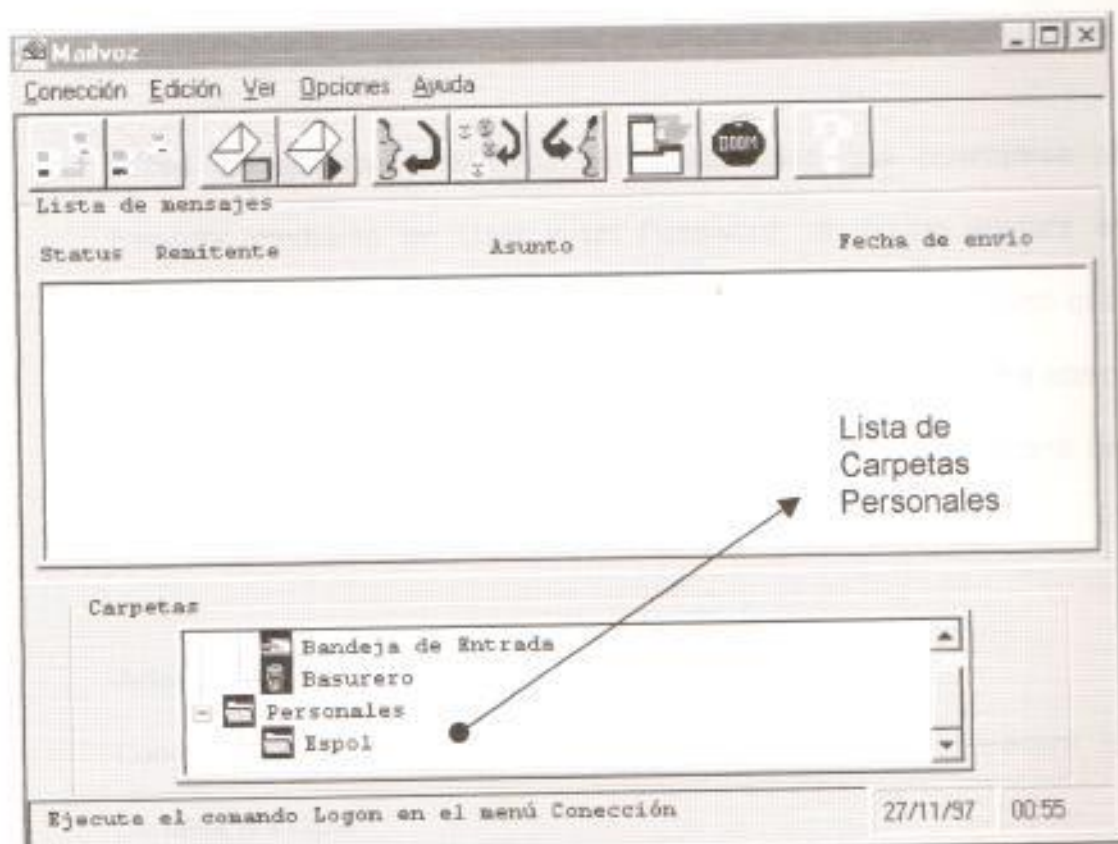


Fig. 38: Pantalla Principal, Lista de Carpetas Personales

Estas carpetas se encuentran en la máquina Cliente. Yo puedo crear las carpetas personales que desee. En la ventana que hemos presentado se ha creado la carpeta Espol.

3.2.- LO PRIMERO QUE TIENE QUE SABER ES:

3.2.1.- ¿Cómo conectarse en red?

Antes de ejecutar el Mailvoz el usuario tiene que conectarse al Servidor mediante un User y un Password, el mismo enviara el mensaje de reconocimiento de que el Password es válido, debido que los usuarios posee una cuenta en el servidor. El Servidor tendrá como sistema operativo Windows NT donde estará instalada la Oficina de Correo que será nuestro

Administrador de Correo.

Luego se ejecuta el programa Mailvoz, donde nos aparecerá la pantalla principal.

3.2.2.- ¿Cuáles son los pasos que se deben seguir para conectarse con el Servidor?

Primer Paso: Para comenzar a establecer la Conexión con el servidor se lo puede hacer de dos maneras:

-En la Barra de Menús se tiene la opción **Conexión** –o mediante teclado Alt+C- y de ahí se elige **Logon** –o mediante teclado Alt+L-.

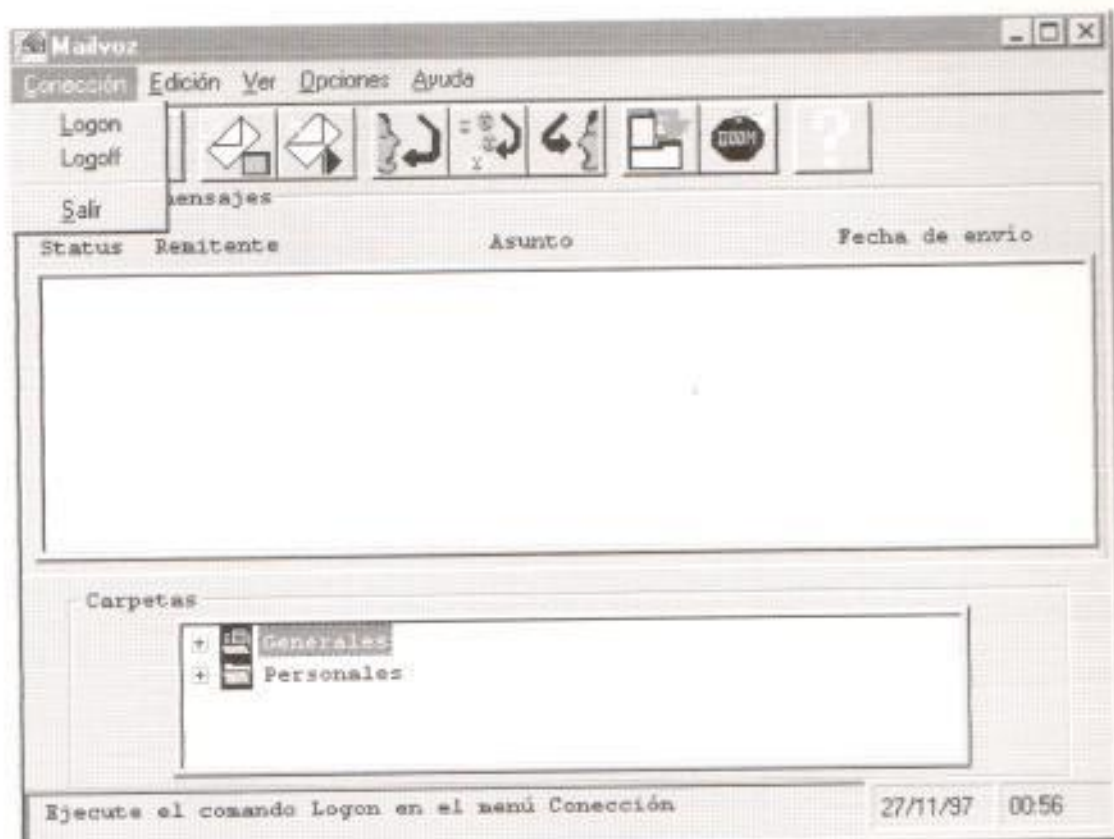


Fig. 39: Pantalla de conexión

- O presionando el siguiente icono:



Fig.40: Icono de conexión

Segundo Paso: Elijiéremos el perfil de usuario con el que vamos a trabajar, por lo tanto nos aparecerá la siguiente pantalla:

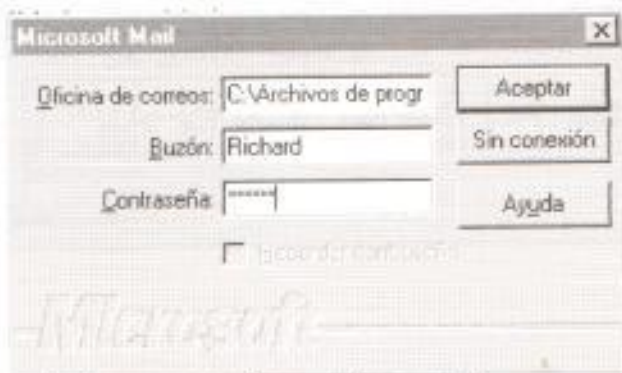


Fig. 42: Ingreso de User y Password

Una vez echo todos estos pasos ya estamos conectados al Servidor.

3.2.3.- ¿Cómo puedo desconectarme del Servidor?

Para desconectarse, se lo puede hacerlo de dos maneras:

- Seleccionamos del menú principal la opción **Conexión** –o mediante teclado Alt+C- y luego **Logoff** –o mediante teclado Alt+G-

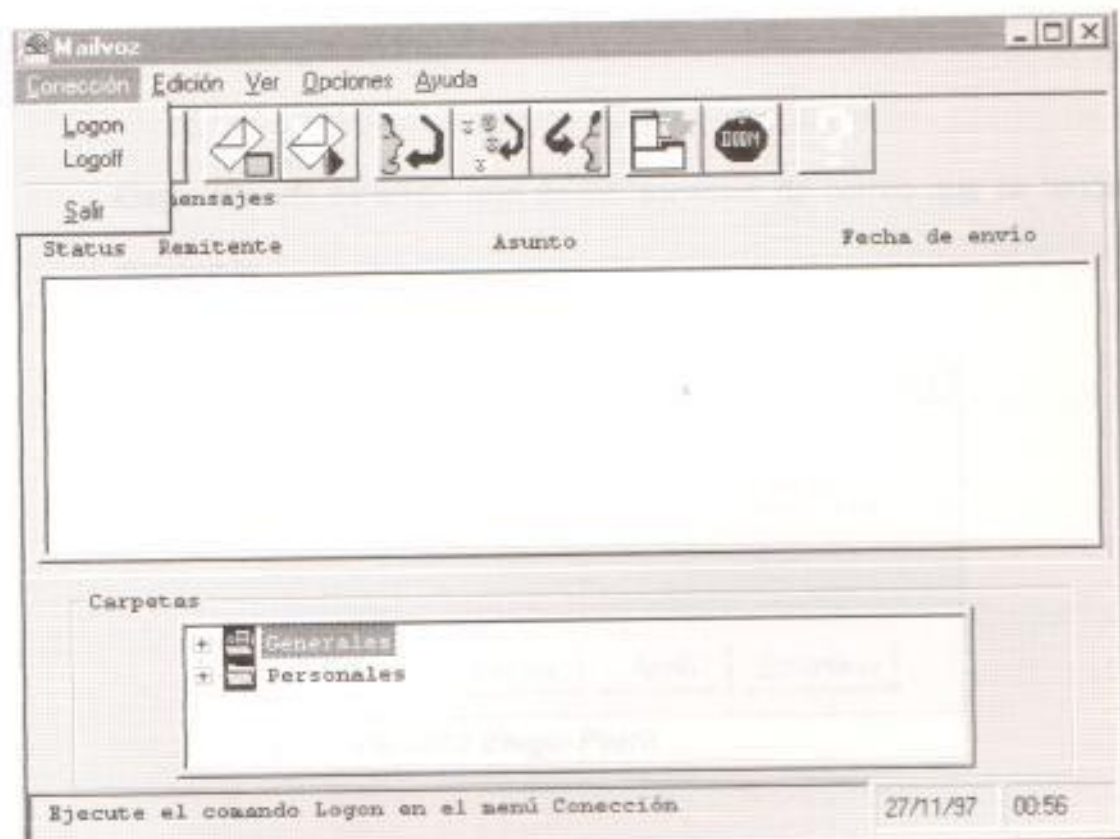


Fig. 43: Pantalla Principal, menú **Conexión**

- O presionando el siguiente icono.



Fig. 44: Icono de cerrar la sesión (Desconectarse)

3.2.4.- ¿Qué es Elegir un Perfil?

Elegir el perfil es elegir uno de los servicios de correo que se tenga disponible.

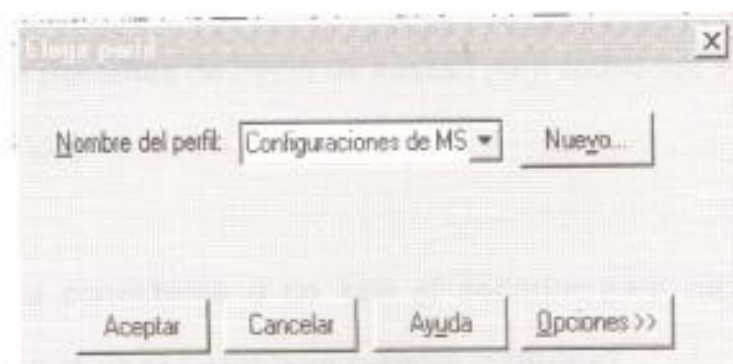


Fig. 45: Pantalla *Elegir Perfil*

Las partes de las cuales está constituida la ventana de *Elegir Perfil* son:

- **[Nombre del perfil]:** Aquí se elige con el perfil que se va a trabajar.
- **Nuevo:** Nos permite establecer un nuevo perfil de correo.
- **Aceptar:** Acepta el perfil elegido.
- **Cancelar:** Cierra la pantalla y no se elige ningún perfil de correo.

-**Ayuda:** Se puede buscar ayuda con respecto a esta ventana.

-**Opciones:** Nos ofrece las siguientes opciones ha elegir:

-{**Configurar como perfil predeterminado**}.

Con está opción elegida no nos volverá a pedir un perfil de correo.

-{**Mostrar pantallas de inicio de sesión para todos los servicios de información**}.

3.2.5.- ¿Se debe conectarse o no con el servidor para reproducir un mensaje que se encuentra en nuestra máquina?

No es necesario conectarse con el Servidor para reproducir el mensaje que se encuentra en nuestra máquina.

3.2.6.- ¿Estando en una sesión se puede establecer una nueva?

Si se puede establecer otra sesión lo que se tiene que hacer es:

- Selecciona del menú principal **Opciones** –o mediante teclado Alt+O-.

- Luego seleccione **Nueva Sesión** –o mediante teclado Alt+N–.

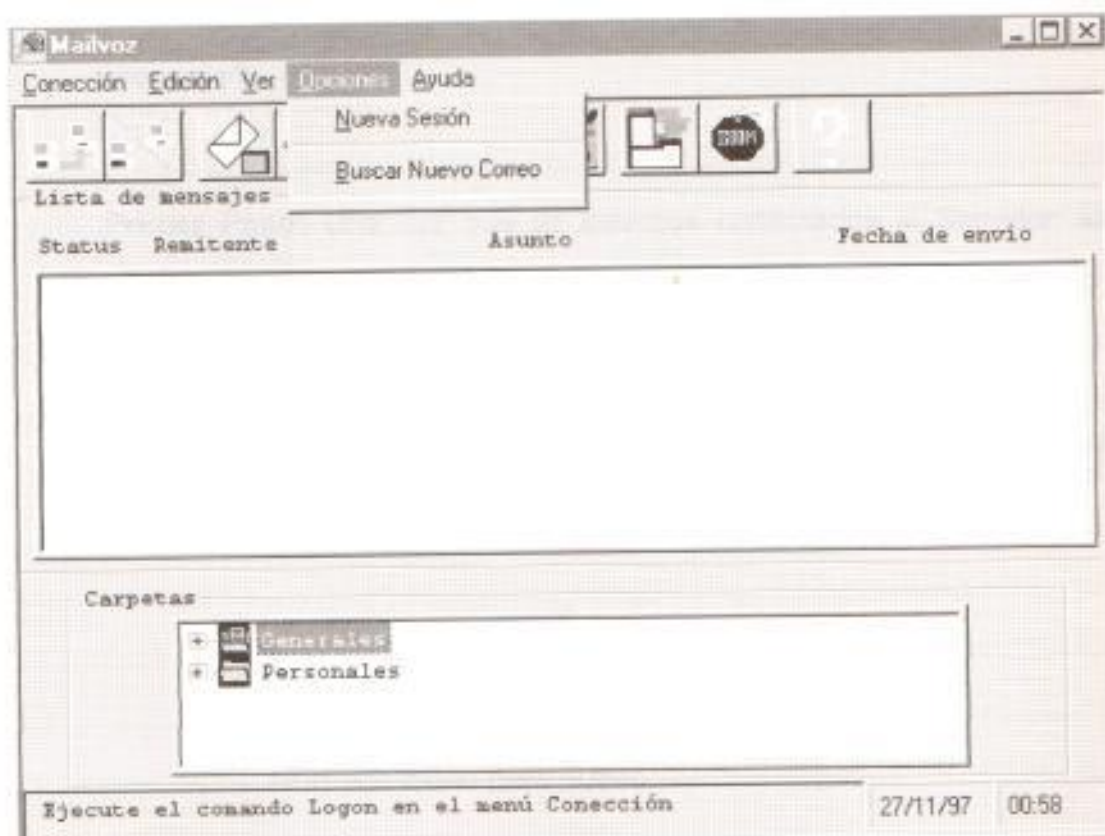


Fig.- 46: Pantalla Principal, menú **Opciones**

Luego nos aparecerá la ventana de **Elegir Perfil**.

3.3.- LO QUE SE DEBE HACER PARA CREAR UN NUEVO MENSAJE.-

3.3.1.- ¿Cuáles son los pasos que se deben seguir para crear un nuevo mensaje?

Primer Paso: Una vez que se estemos conectados al Servidor, se puede comenzar de dos maneras:

-Seleccionando el siguiente icono que se encuentra en su respectiva barra:



Fig. 47: Icono para nuevo

- O en el menú principal seleccionamos **Edición** –o mediante teclado Alt+E- y elegimos la opción **Nuevo** –o mediante teclado Alt+N-.

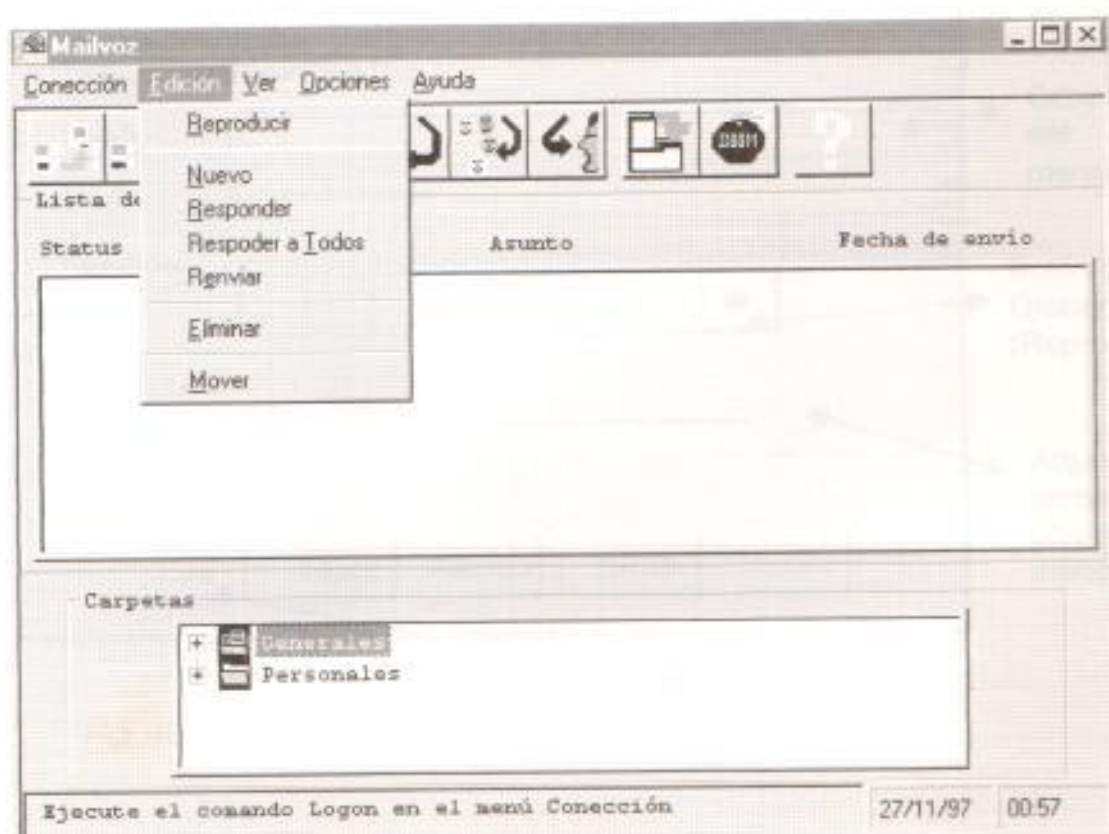


Fig. 48: Pantalla Principal, menú **Edición**

Después de esto nos aparecerá la ventana de **Reproducir Mensaje**.

Esta ventana consta de las siguientes partes:

- Un recuadro que lleva como título **<Cabecera del Mensaje>**.
- El **<Reproductor>** que nos ayudará a grabar el mensaje.
- El recuadro de **<Archivos Adjuntos>** se colocará información sobre los archivos que se enviarán junto con el mensaje.

-Botones como Enviar, Adjuntar, Comprobar, Dirección, Cancelar

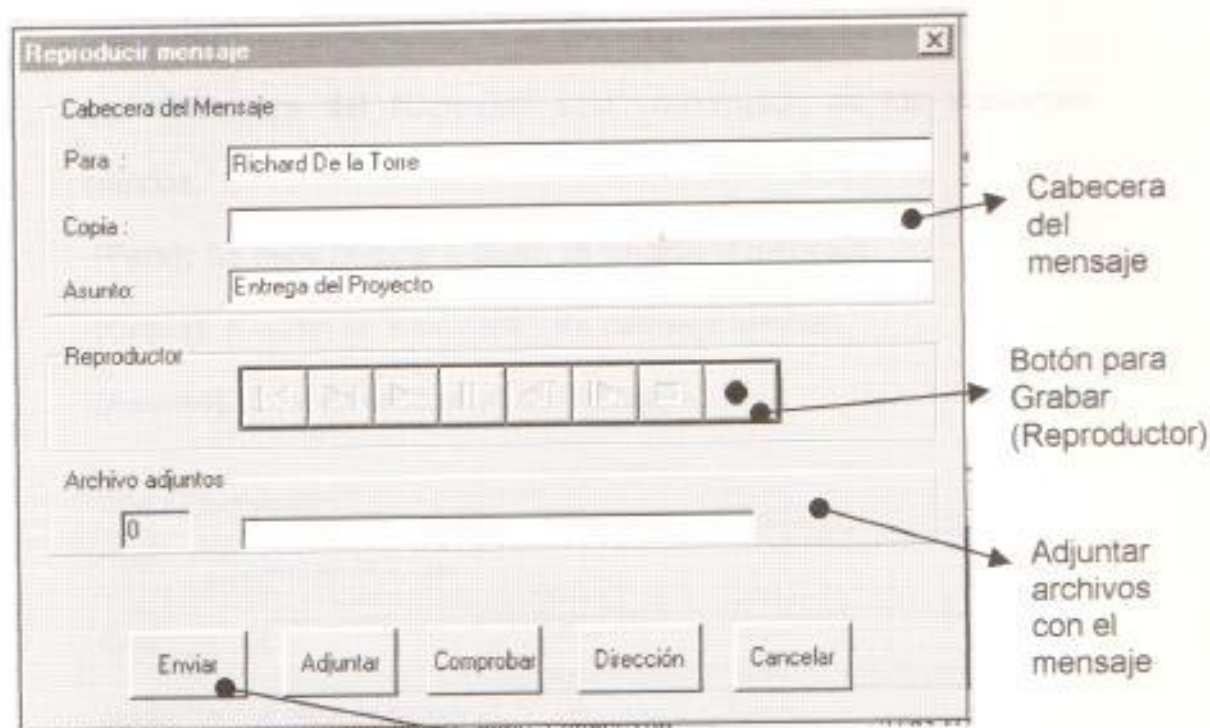


Fig. 49: Pantalla **Reproducir Mensaje**

Se envía el mensaje una vez grabado

Segundo Paso: Escribir la <Cabecera del Mensaje>, se llena los campos que contiene.

Tercer Paso: Se comienza a grabar los mensajes, presionando el botón de Grabar que se encuentra en el <Reproductor>:

Cuarto Paso: Por último se envía el mensaje presionando el botón Enviar.

3.3.2.-¿Qué se debe colocar en la Cabecera del mensaje (Enviar mensaje)?

La <Cabecera del Mensaje> está conformada de los siguientes campos:

-[Para]: Se debe colocar a quién va dirigida el mensaje.

-[Copia]: A quién se le enviará este mensaje también.

-[Asunto]: Cual es el objetivo del mensaje.



Cabecera del Mensaje	
Para :	Richard De la Torre
Copia :	
Asunto:	Entrega del Proyecto

Fig. 50: Pantalla *Reproducir mensaje*, <Cabecera del Mensaje>

3.3.3.-¿Cómo obtener la dirección de los usuarios que tienen buzones (cuentas) en el Servidor?

Para obtener dicha información se tiene que presionar el botón

Dirección

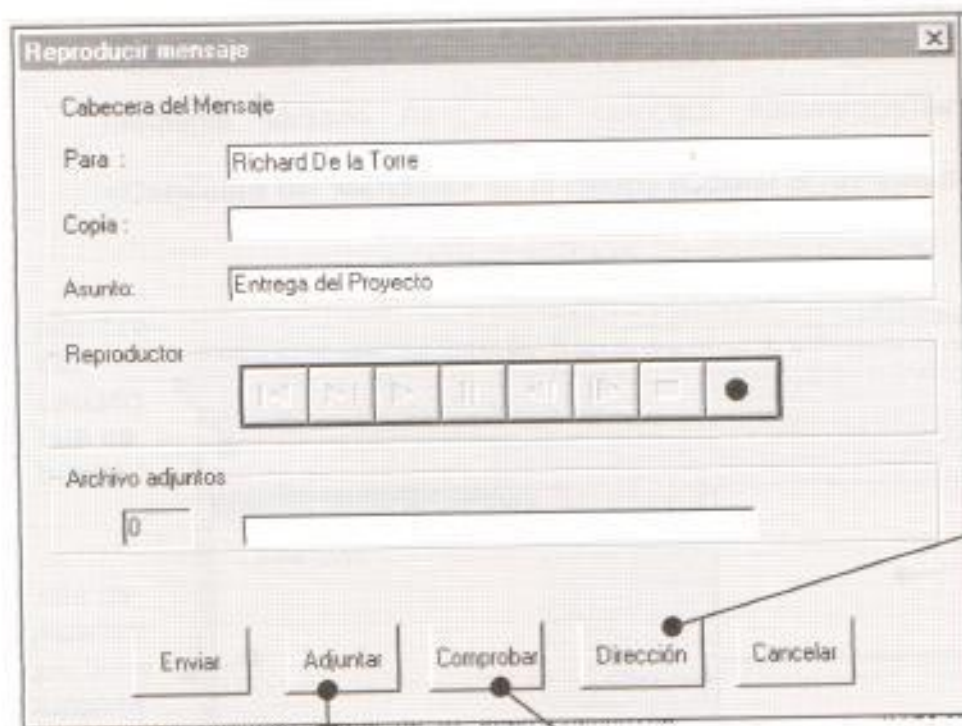


Fig. 51: Pantalla *Reproducir Mensaje*

Después de esto nos aparecerá la pantalla de la Libreta de Direcciones, que tiene las siguientes partes:

- [Mostrar los nombres que figuran en]:** Se indicaran donde se debe buscar las direcciones.
- [Escriba el nombre o selecciónelo en la lista]:** Se escribe el nombre del usuario.

-**Para->**: Se puede seleccionar un usuario y presionando dicho botón – o mediante teclado Alt+P-, se colocará automáticamente en la <Cabecera del Mensaje> en el campo **[Para]** el nombre del usuario.

-**Cc->**: Se puede seleccionar un usuario y presionando dicho botón – o mediante teclado Alt+C-, se colocará automáticamente en la <Cabecera del Mensaje> en el campo **[Copia]** el nombre del usuario.

En que oficina de correo
se quiere buscar

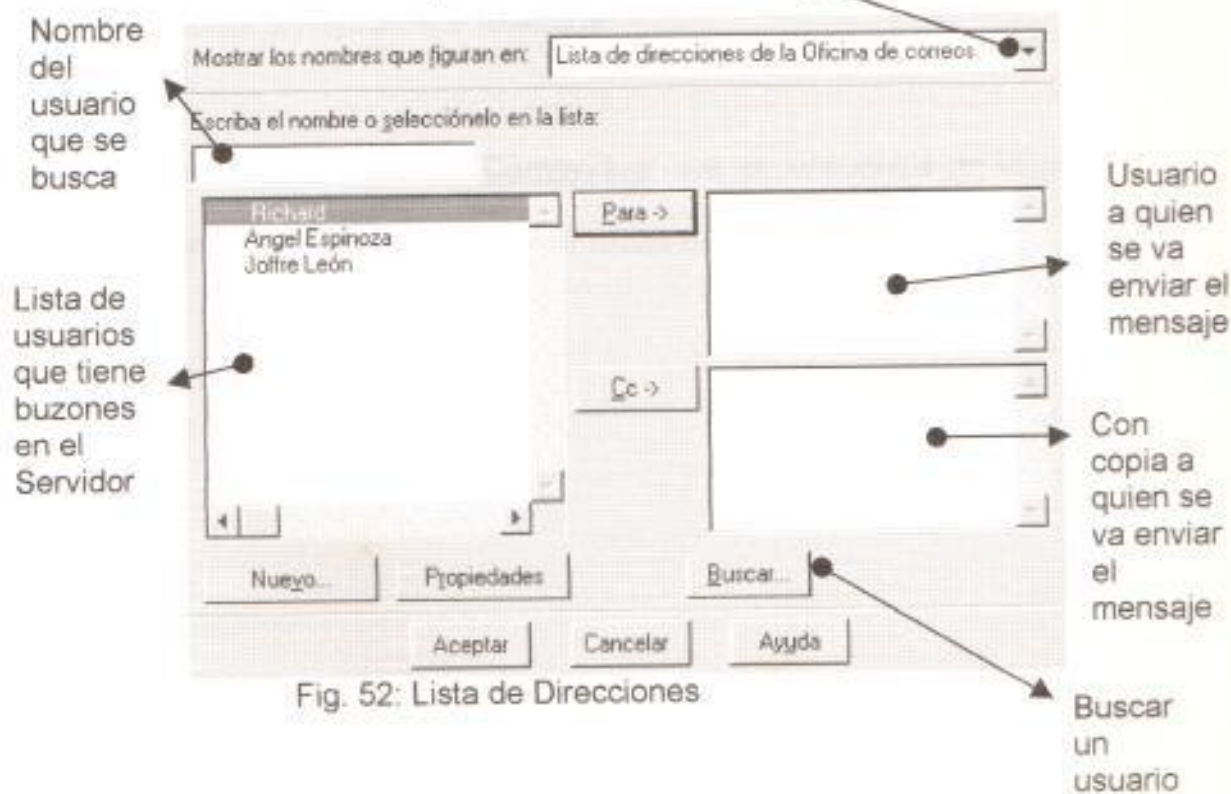


Fig. 52: Lista de Direcciones

- **Buscar...**: Nos ayuda a buscar un usuario en la Lista de Direcciones del correo.

- **Aceptar**: Acepta lo elegido.

- **Cancelar**: Cancela lo elegido.

- Ayuda: Nos lleva al menú de ayuda.

3.3.4.-¿Se puede comprobar el nombre del usuario a quien yo quiero enviarle el mensaje?

Estando activa la pantalla **Reproducir Mensaje**, usted puede comprobar el nombre del usuario de dos maneras:

-Presionando el botón Dirección que se encuentra en la parte inferior de la ventana **Reproducir mensaje**.

-O presionando el botón Comprobar que se encuentra en la parte inferior de la pantalla **Reproducir mensaje**.

3.3.5.- ¿Cómo manejo el Reproductor para grabar un mensaje?

La carta se puede comenzar a grabar una vez que se ha llenado los campos que se encuentran en la <cabecera del mensaje>.

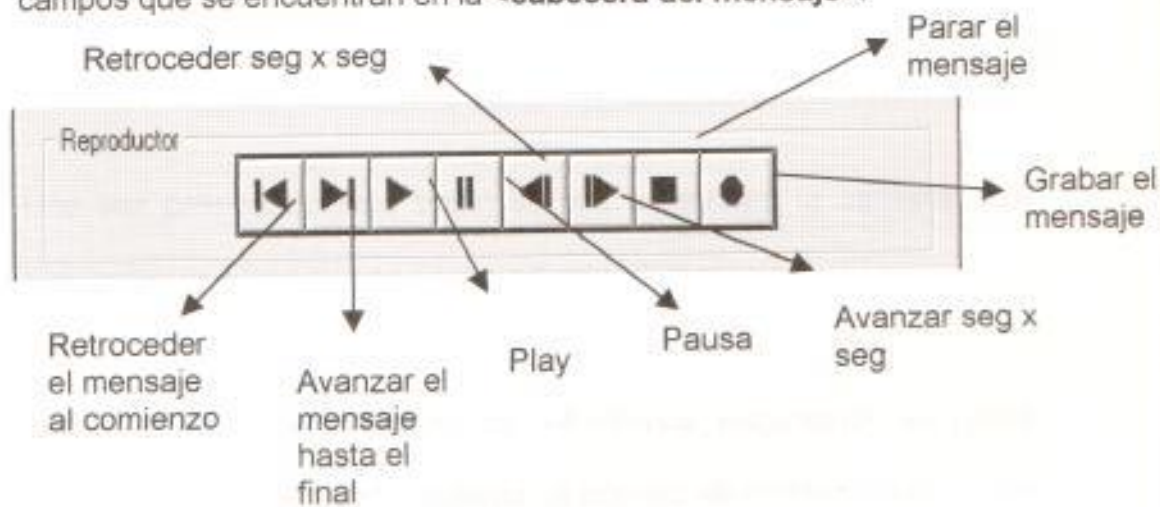


Fig. 53: Reproductor de *Reproducir Mensaje*

Luego se presiona el botón **Grabar** y se comienza a hablar.

El Reproductor tiene otros botones que sirven para:

- **Retroceder** el mensaje hasta el inicio del mismo.
- **Avanzar** el mensaje hasta el final del mismo.
- Realizar un **Play** al mensaje.
- Hacer una **Pausa** a la carta cuando se la está grabando.
- **Retroceder segundo por segundo** el mensaje.
- **Avanzar segundo por segundo** el mensaje.
- **Parar** la grabación del mensaje.

3.3.6.-¿Cómo se puede adjuntar archivos para ser enviados con el mensaje?

Después que se ha llenado la cabecera y/o grabado el mensaje, podemos adjuntar archivos al mismo presionando el botón **Adjuntar**.

Una vez presionado este botón se nos presentará la ventana para buscar los archivos que se desee adjuntar al mensaje.

Por lo tanto en el recuadro de **<Archivos Adjuntos>** se podrá observar: En el recuadro izquierdo, el número de archivos que se van adjuntando y en el otro recuadro el nombre de cada uno de los archivos que se adjuntan.

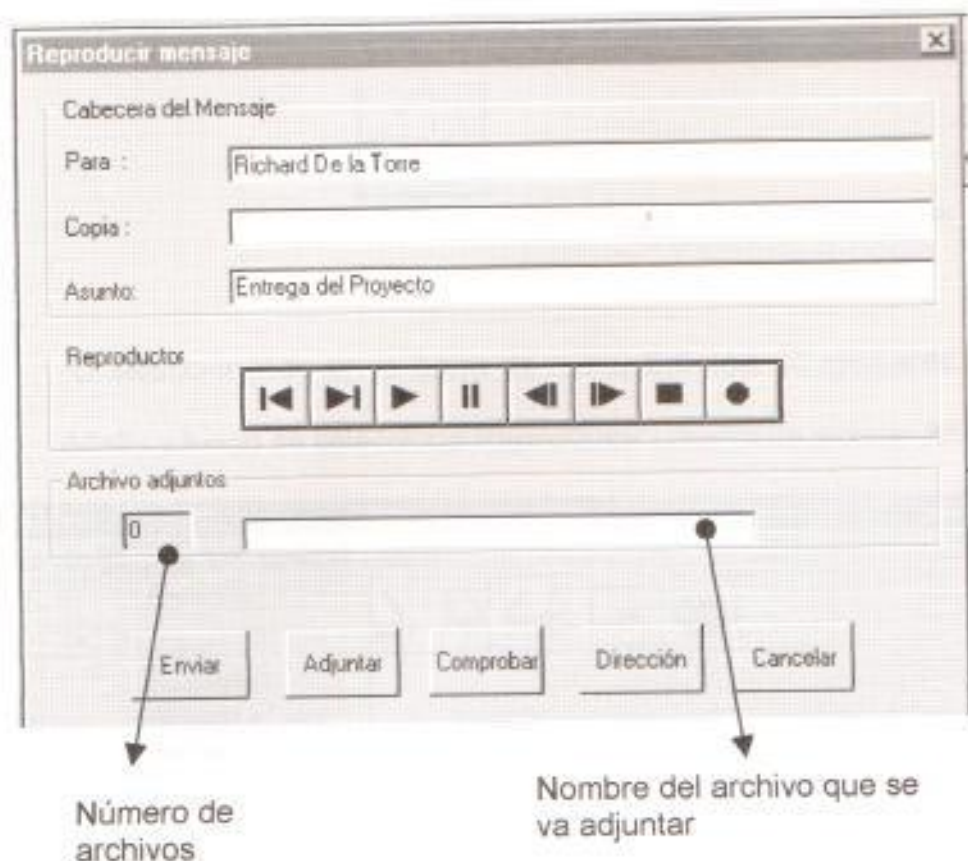


Fig. 54: Pantalla *Reproducir Mensaje*, Archivos adjuntos

3.4.- LO QUE SE DEBE HACER PARA LEER UN MENSAJE.-

3.4.1.- ¿Cuáles son los pasos para Reproducir un Mensaje?

Primer Paso: Seleccionar el mensaje que se desea escuchar desde la <Lista de Mensajes> en la Pantalla Principal.

Segundo Paso: Hay dos maneras para comenzar a reproducir un mensaje:

- Presionando el icono mostrado a continuación:



Fig. 55: Icono para Leer un Mensaje

-O seleccionando en la barra de menú **Edición** –o mediante teclado **Alt+E-** y luego se selecciona **Reproducir** –mediante teclado **Alt+R-**.

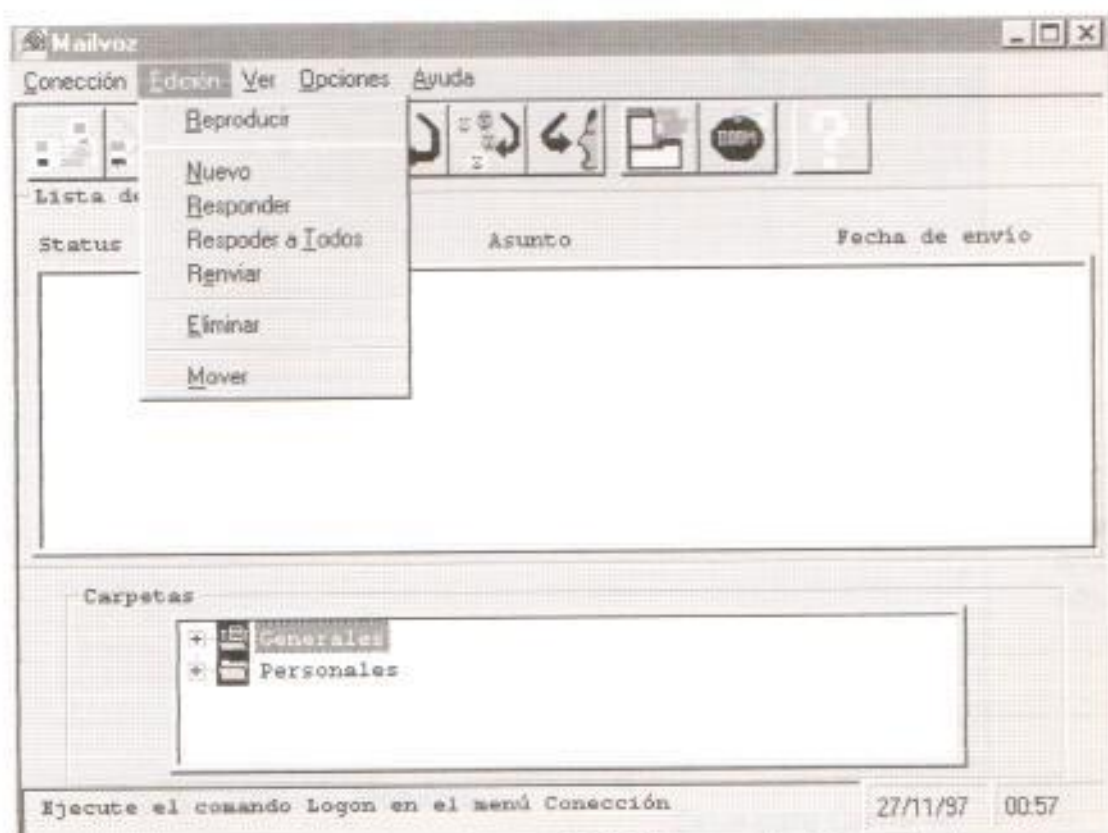


Fig. 56: Pantalla Principal, menú **Edición**, opción **Reproducir**

Luego se nos presentara la pantalla **Mensaje**. Dicha pantalla consta de las siguientes partes:

- Un recuadro que lleva como titulo **<Cabecera del Mensaje>**.
- El **<Reproductor>** que nos ayudará a leer el mensaje.
- El recuadro de **<Archivos Adjuntos>** tiene información sobre los archivos que se han enviado junto con el mensaje.
- Un Botón que es **Cerrar mensaje**, cancela la ventana o cierra el mensaje que se está leyendo.

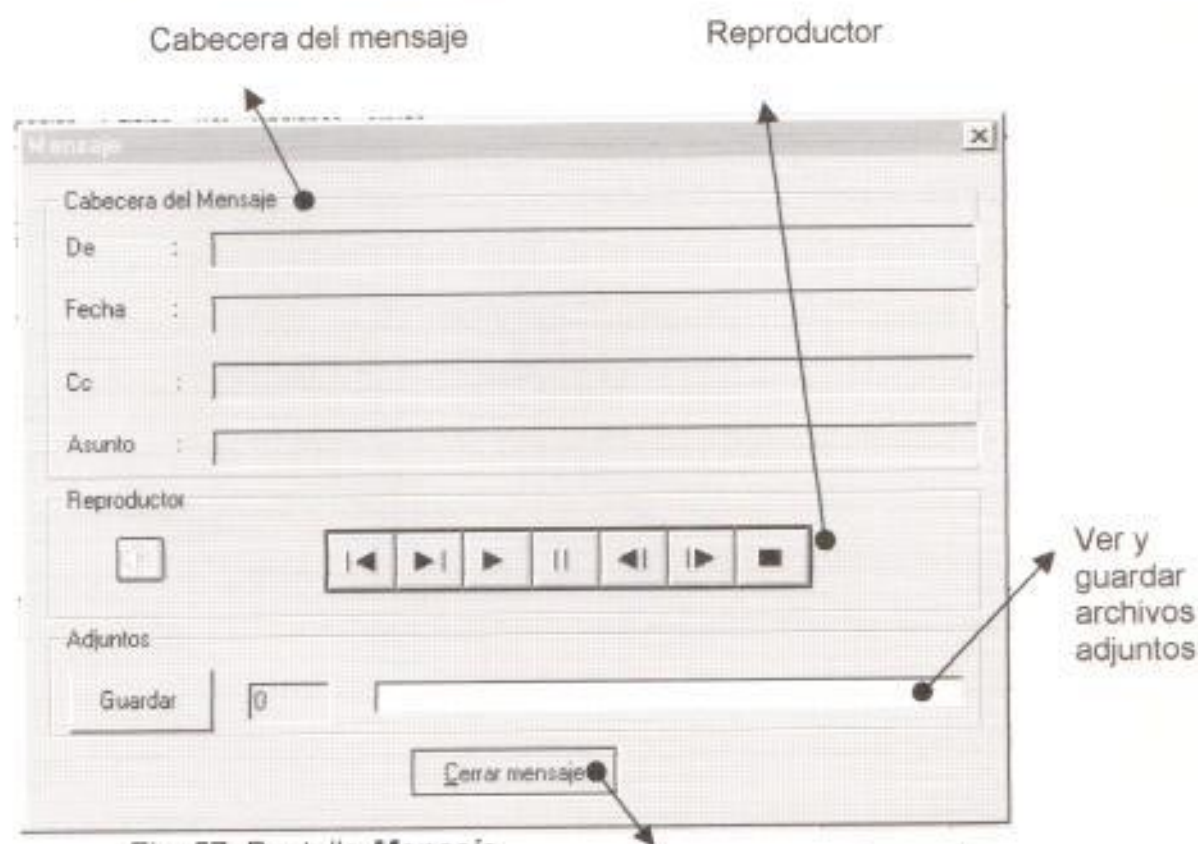


Fig. 57: Pantalla **Mensaje**

Sirve para Cerrar el mensaje

Tercer Paso: Presionar el botón de **Play** para comenzar a escuchar el mensaje.

Cuarto Paso: Si se desea cerrar el mensaje se presiona el botón **Cerra mensaje** –o mediante teclado Alt+C-.

3.4.2.-¿Qué nos muestra la Cabecera del mensaje de la pantalla Mensaje (Reproducir un Mensaje)?

La <Cabecera del mensaje> nos da información sobre:

Cabecera del Mensaje	
De :	<input type="text"/>
Fecha :	<input type="text"/>
Cc :	<input type="text"/>
Asunto :	<input type="text"/>

Fig. 58: Cabecera del Mensaje en la Pantalla Mensaje

-**De:** Nos indicará quién nos envió la carta.

-**Fecha:** Nos muestra en que fecha nos enviaron el mensaje.

-**Cc:** Esto significa con copia, nos indicará a quién más le enviaron este mensaje.

-**Asunto:** De que se está tratando el mensaje.

3.4.3.- ¿El Reproductor para reproducir mensajes se maneja igual que el Reproductor para enviar mensajes?

Si se manejan iguales, solo que el reproductor que aparece en la pantalla **Mensaje** no tiene la opción de **Grabar**.

A continuación se tiene al Reproductor de Reproducir mensaje:

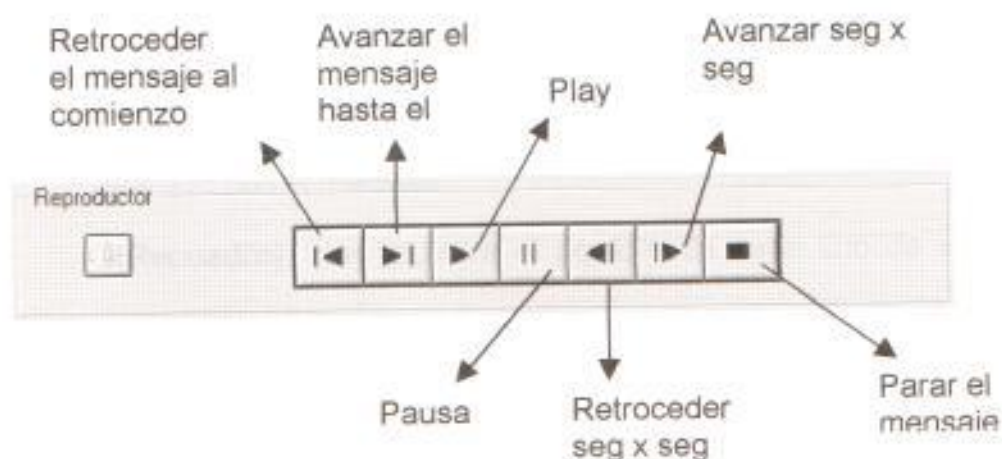


Fig. 59: Reproductor de la Pantalla **Mensaje**

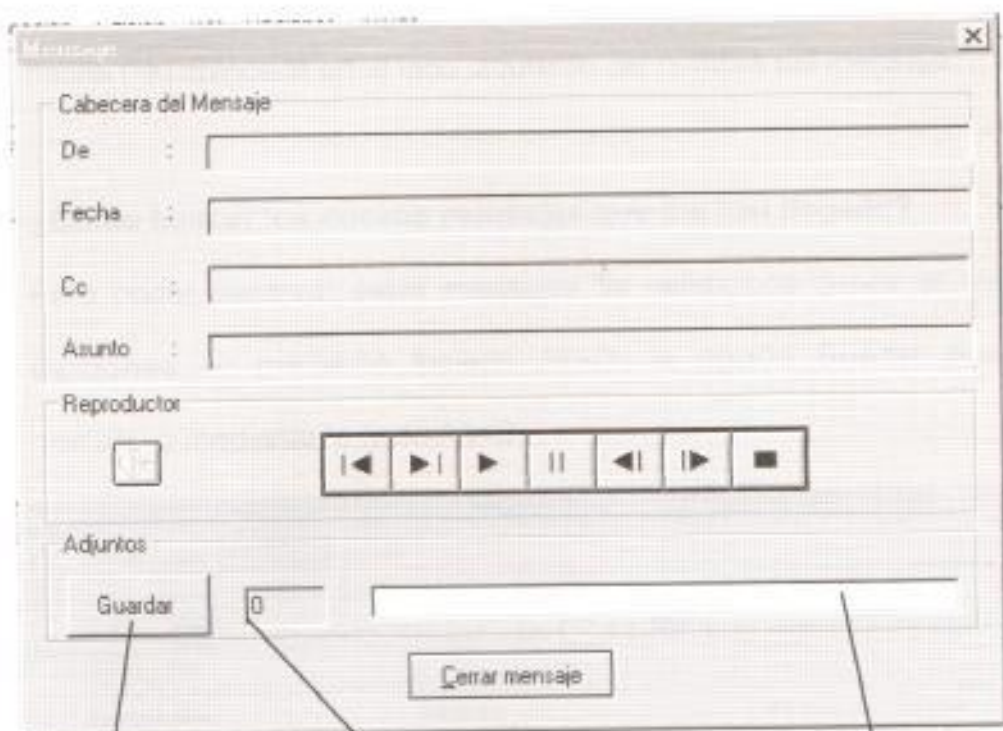
3.4.4.-¿Cómo saber si se han enviado archivos adjuntos con el mensaje?

Los archivos que se han adjuntado al mensaje se los muestra en el recuadro de la pantalla **Mensaje** que tiene como título <Adjuntos> y consta de lo siguientes:

-**Guardar**: Podemos guardar los archivos que nos han adjuntado al mensaje, en cualquier directorio, carpeta que deseáramos.

-**Cuadro con número**: Nos indica cuantos archivos adjuntos nos han enviado con el mensaje

-**[Recuadro]**: Nos muestra el nombre de cada uno de los archivos adjuntos que han sido enviados con el mensaje.



Nos permite guardar el archivo en cualquier carpeta

Cuantos archivos estan adjuntados al mensaje

Nombre del archivo adjunto

Fig. 60: Pantalla **Mensaje**, Archivos Adjuntos

3.4.5.- ¿Cómo reconocer que un mensaje es nuevo?

Al mensaje nuevo se lo reconoce porque se puede observar una flecha (→) colocada en el lado izquierdo del nombre del mensaje.

3.4.6.- ¿Cómo buscar los nuevos mensajes que me han llegado?

Para poder observar estos mensajes se selecciona desde el menú **Opciones** -o mediante teclado Alt+O- la opción **Buscar nuevo correo** -o mediante teclado Alt+B-

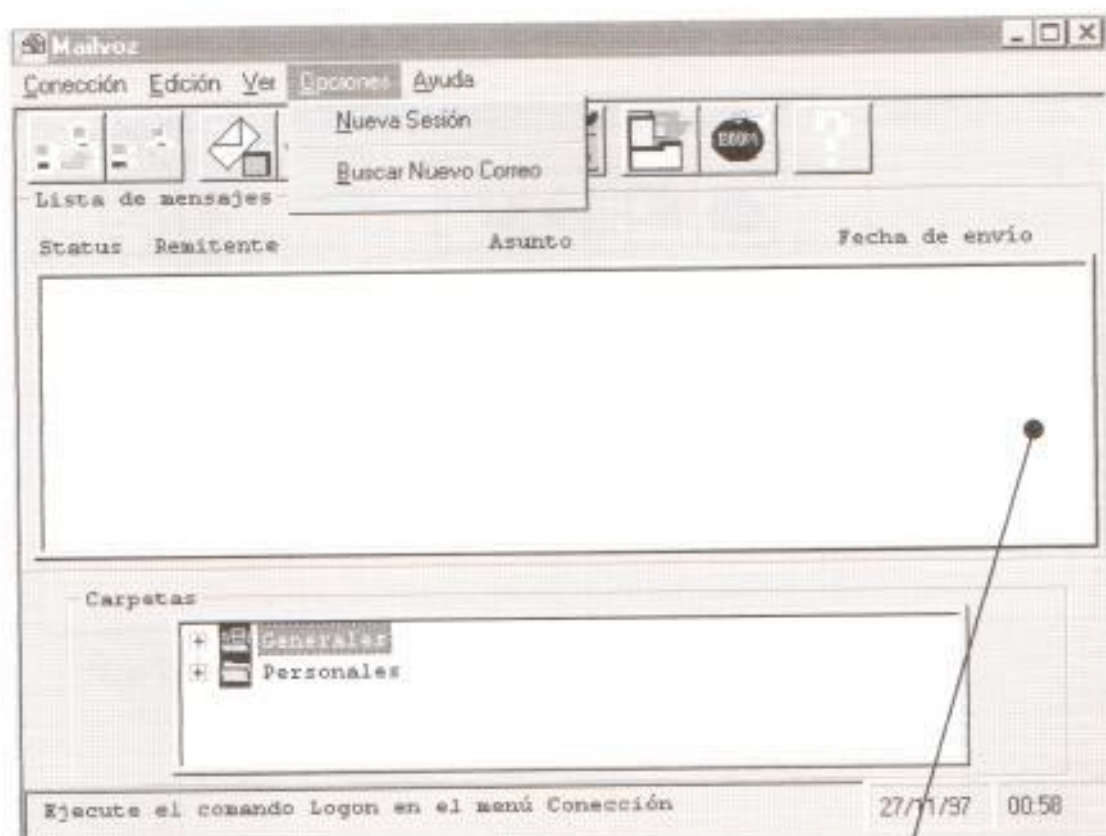


Fig. 61: Pantalla Principal, menú **Opciones**

Listas de mensajes

Luego de esto nos aparecerá en la <Listas de mensajes> todos los mensajes que son nuevos (que no han sido leídos).

3.4.7.- ¿Cómo poder ver los mensajes no leídos?

Para poder observar estos mensajes, elegimos la opción **ver** del menú –o mediante teclado Alt+V- y luego se elige **No leídos** –o mediante teclado Alt+N-. Después de esto los mensajes nos aparecerá en la <Lista de mensajes>

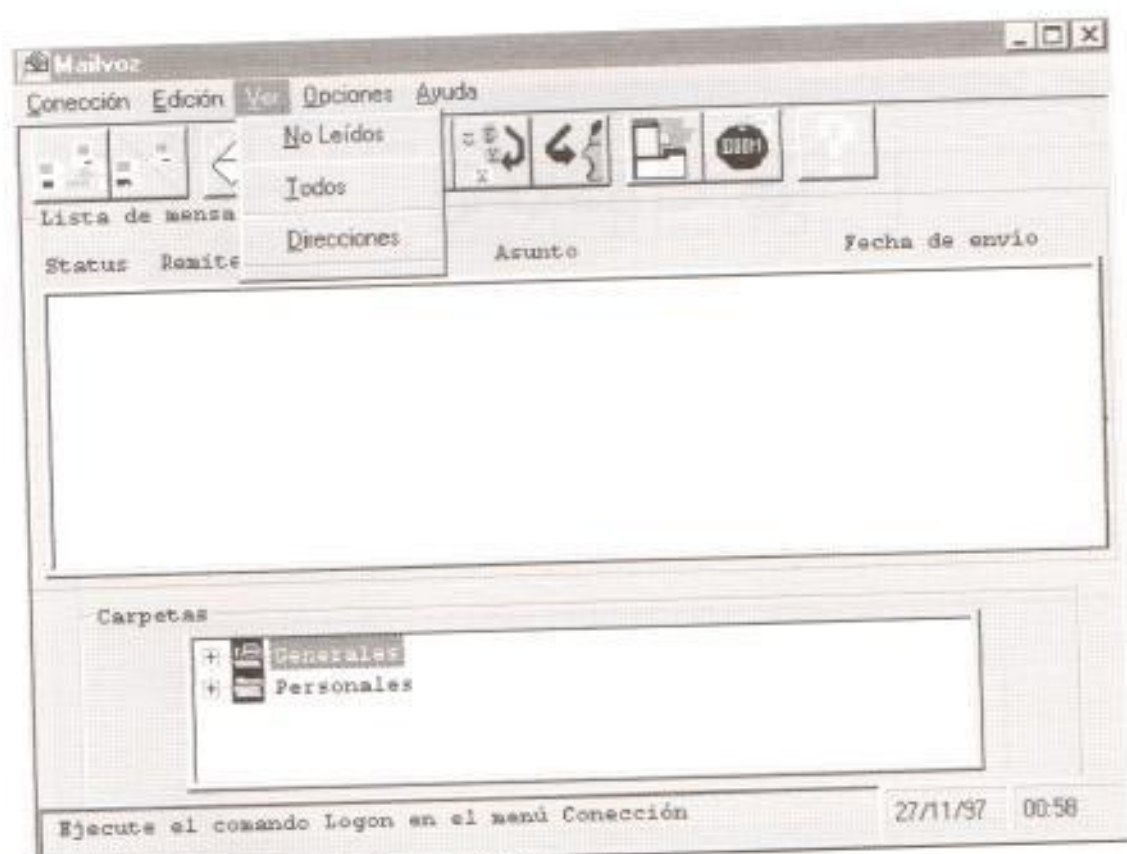


Fig. 62: Pantalla Principal, menú Ver

3.5.- LO QUE SE DEBE HACER PARA RESPONDER LOS MENSAJES.-

3.5.1.-Pasos para responder un mensaje.-

Los pasos a seguir son:

Primer Paso: Seleccionamos un mensaje que se encuentre en la <Lista de mensajes>.

Segundo paso: En el menú principal se selecciona **Edición** —o mediante teclado Alt+E— y después se selecciona **Responder** —o mediante teclado Alt+R—.

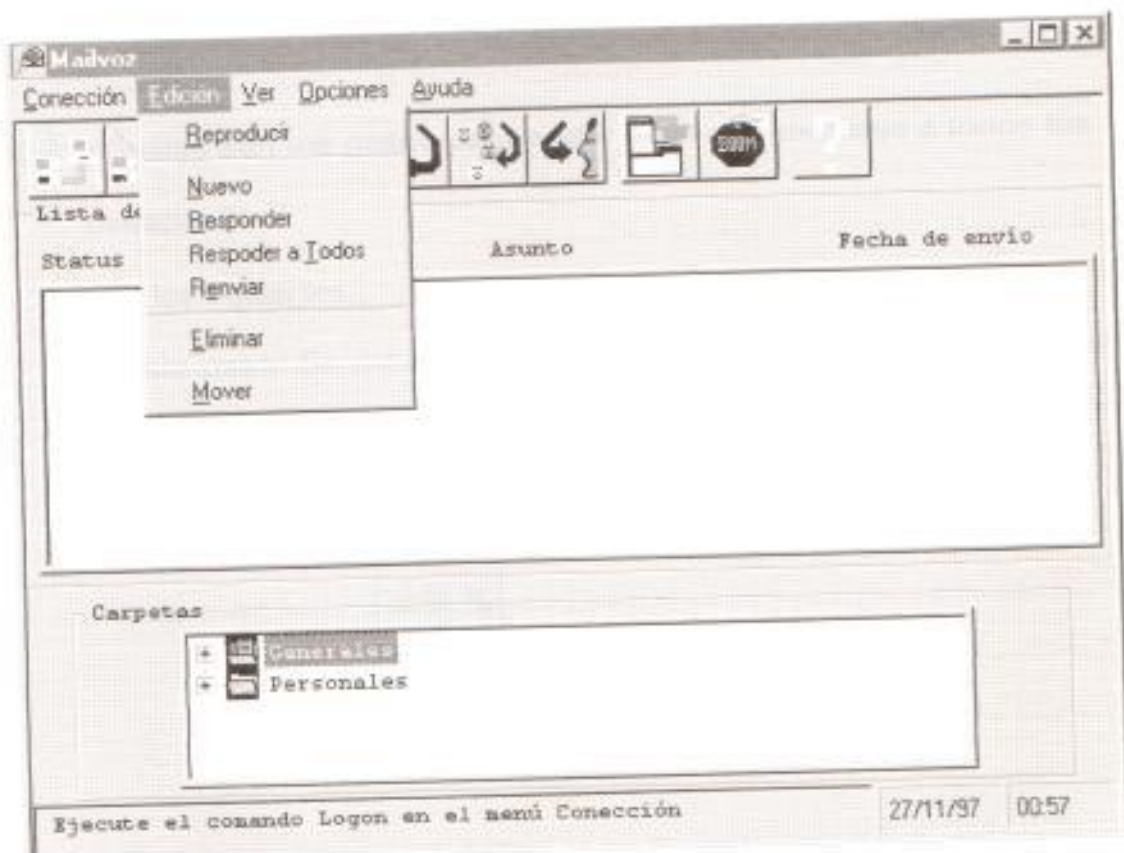


Fig. 63: Pantalla Principal, menú **Edición** se escoge la opción **Responder**

Tercer paso: Se presentará la pantalla *Reproducir mensaje*, pero en la **<Cabecera del Mensaje>** ya no se escribirán sus campos ya que automáticamente se cargan con los datos del mensaje seleccionado. (Ver Fig.)

Cuarto Paso: Se comienza a grabar el mensaje presionando el botón **Grabar**. Se puede también adjuntar al mensaje archivos. (Ver Fig.)

Quinto Paso: Se presiona el botón **Enviar**.

3.5.2.- ¿Cuáles son los pasos para poder enviar un mensaje a todos los usuarios?

Son los siguientes:

Primer paso: Se debe seleccionar la opción de **Responder a todos** los usuarios y se lo hace de dos maneras:

- Seleccionando el siguiente icono:



Fig. 64: Icono para Responder a todos los usuarios

- O eligiendo la opción **Edición** –o mediante teclado Alt+E- del menú y luego se elige **Responder a Todos** –o mediante teclado Alt+T-.



Fig. 65: Pantalla Principal, menú **Edición** se elige **Responder a Todos**.

Segundo Paso: Nos aparece la pantalla *Reproducir mensaje*, pero ya tendrá llenado todos los campos en la <Cabecera del Mensaje>.

Tercer Paso: Se comienza a grabar el mensaje presionando el botón **Grabar**. Se puede también adjuntar al mensaje archivos. (Ver Fig.)

Cuarto Paso: Se presiona el botón **Enviar**.

3.6.- LO QUE SE DEBE HACER PARA REENVIAR UN MENSAJE.-

3.6.1.- ¿Cómo se puede reenviar un mensaje recibido?

Los pasos para reenviar un mensaje son:

Primer Paso: Seleccionamos el mensaje que se va a reenviar.

Segundo Paso: Se puede comenzar de dos formas:

-En la barra de menú seleccionamos **Edición** -o mediante teclado Alt+E- y luego seleccionamos **Reenviar** -o mediante teclado Alt+E-.

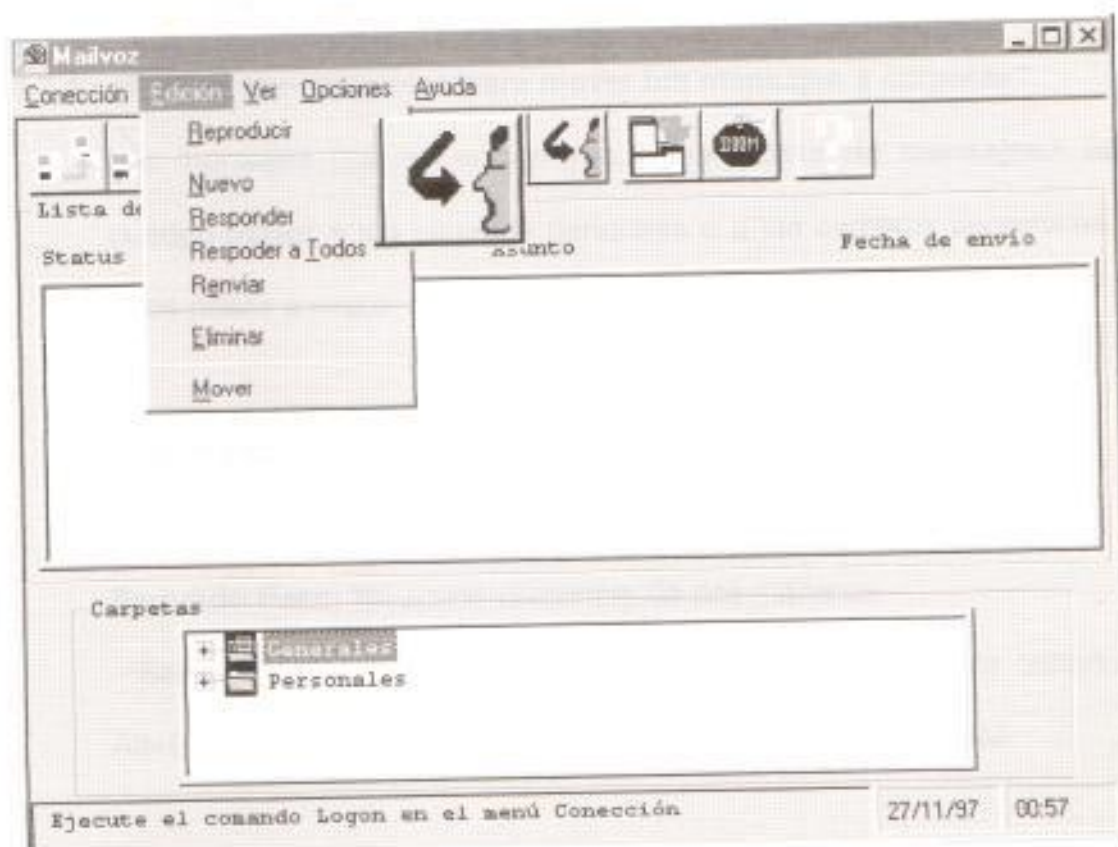


Fig. 66: Pantalla Principal, menú **Edición** se elige **Reenviar**

O mediante el icono de Reenviar.

Tercer Paso: Se nos presentará la pantalla de **Reproducir Mensaje**. En la cual se llenara: los campos que se encuentran en la **<Cabecera del Mensaje>**, se puede también adjuntar archivos al mensaje si desea.

Cuarto Paso: Se presiona el botón **Enviar**.

3.7.- LO QUE SE DEBE HACER PARA MOVER MENSAJES.-

3.7.1.- ¿Cuáles son los pasos para mover los mensajes a carpetas?

Los mensajes que se encuentran en la **<Lista de mensajes>** se pueden mover a las carpetas generales o a las carpetas personales.

Y los pasos a seguir son:

Primer Paso: Se selecciona un mensaje de la **<Lista de mensajes>**.

Segundo Paso: Se puede comenzar de dos maneras:

- Seleccionamos en el menú principal **Edición** –o mediante teclado

Alt+E- y luego se selecciona **Mover** –o mediante teclado Alt+M-

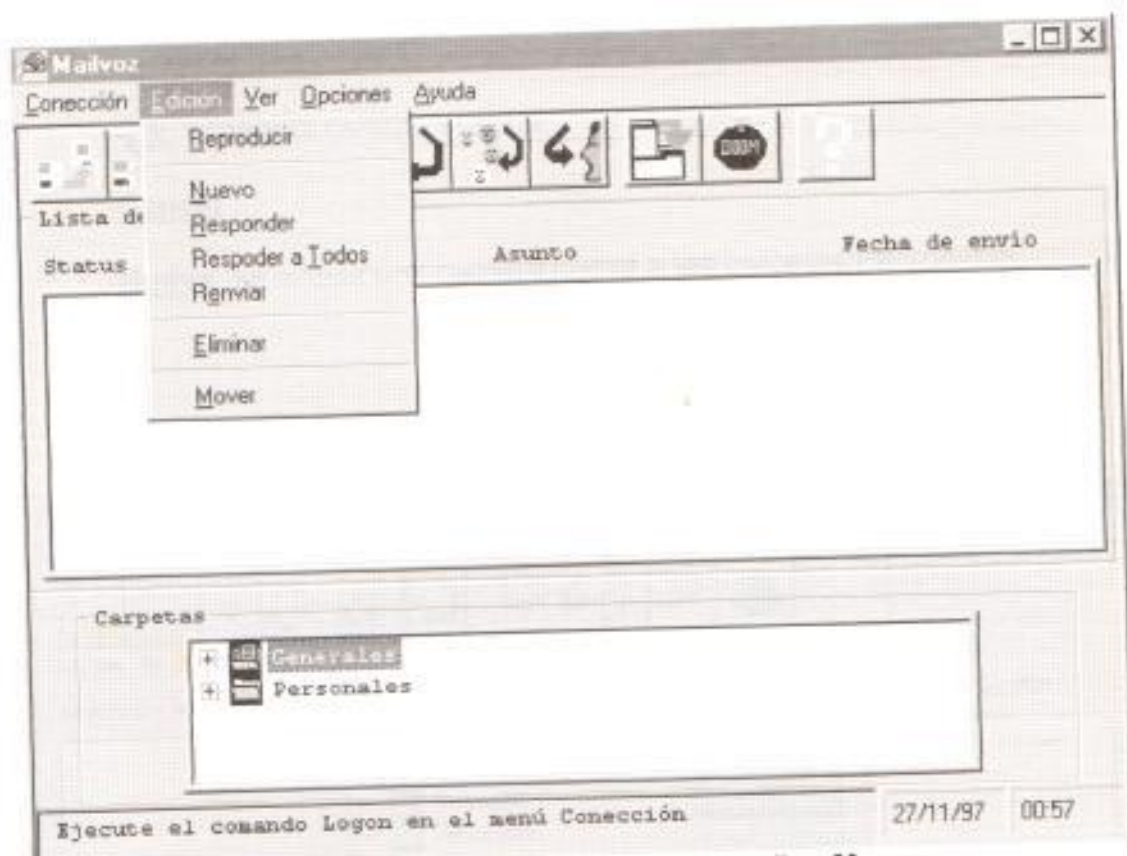


Fig. 67: Pantalla principal, menú **Edición** se elige **Mover**

- También se puede presionar el siguiente icono:



Fig. 68: Icono para mover mensajes a carpetas

Tercer Paso: Nos aparecerá la pantalla **Mover a una carpeta**. En la cual observaremos las dos carpetas que tenemos en un principio: Generales y Personales. También se observará los siguientes botones:

- Ok, acepta la carpeta elegida.

- Cancel

- Nueva, se añade una carpeta nueva a las carpetas Personales.

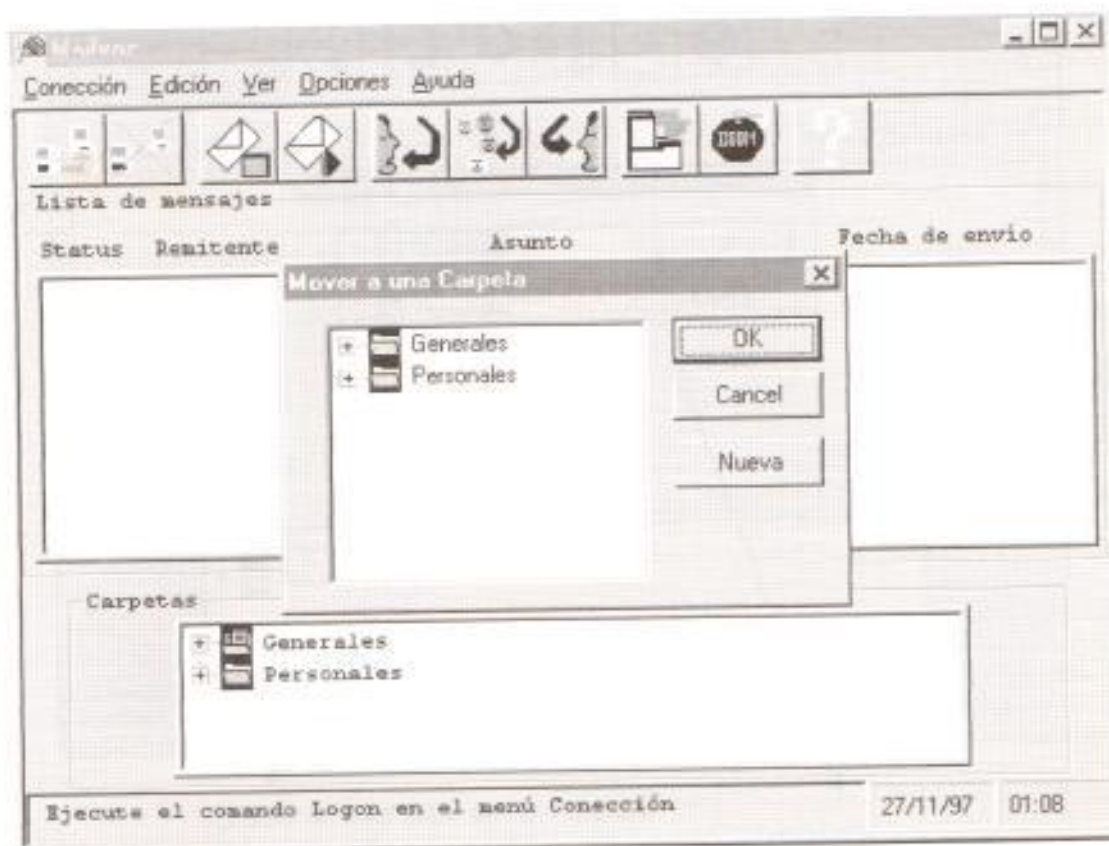


Fig. 69: Pantalla *Mover una carpeta*

Cuarto Paso: Si se desea se puede añadir carpetas a las carpetas Personales, esto se lo hace presionando el botón **Nueva**.

Dentro de las carpetas Generales tenemos a la carpeta Basurero.

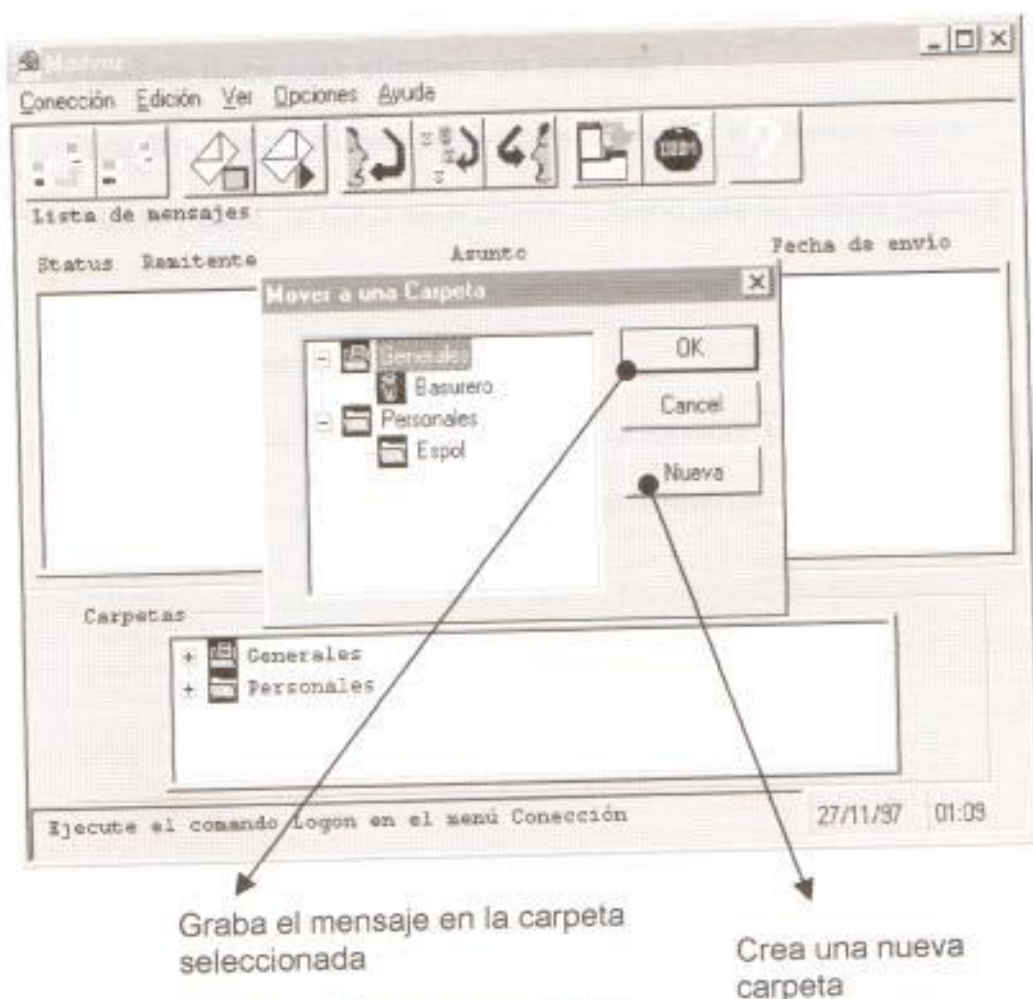


Fig. 70: Pantalla **Mover a una carpeta**

Quinto Paso: Una vez seleccionada la dirección en la cual se desea grabar el mensaje se presiona **OK**.

3.8.- LO QUE SE DEBE HACER PARA ELIMINAR UN MENSAJE.-

3.8.1.- ¿Qué se hace para eliminar un mensaje?

Para eliminar los mensajes se debe seguir los siguientes pasos:

Primer Paso: Se debe seleccionar el menú que se desea eliminar.

Segundo Paso:

-Se selecciona del Menú Principal la opción **Edición** –o mediante teclado **Alt+E-** y luego seleccionamos **Eliminar** –o mediante teclado **Alt+E-**.

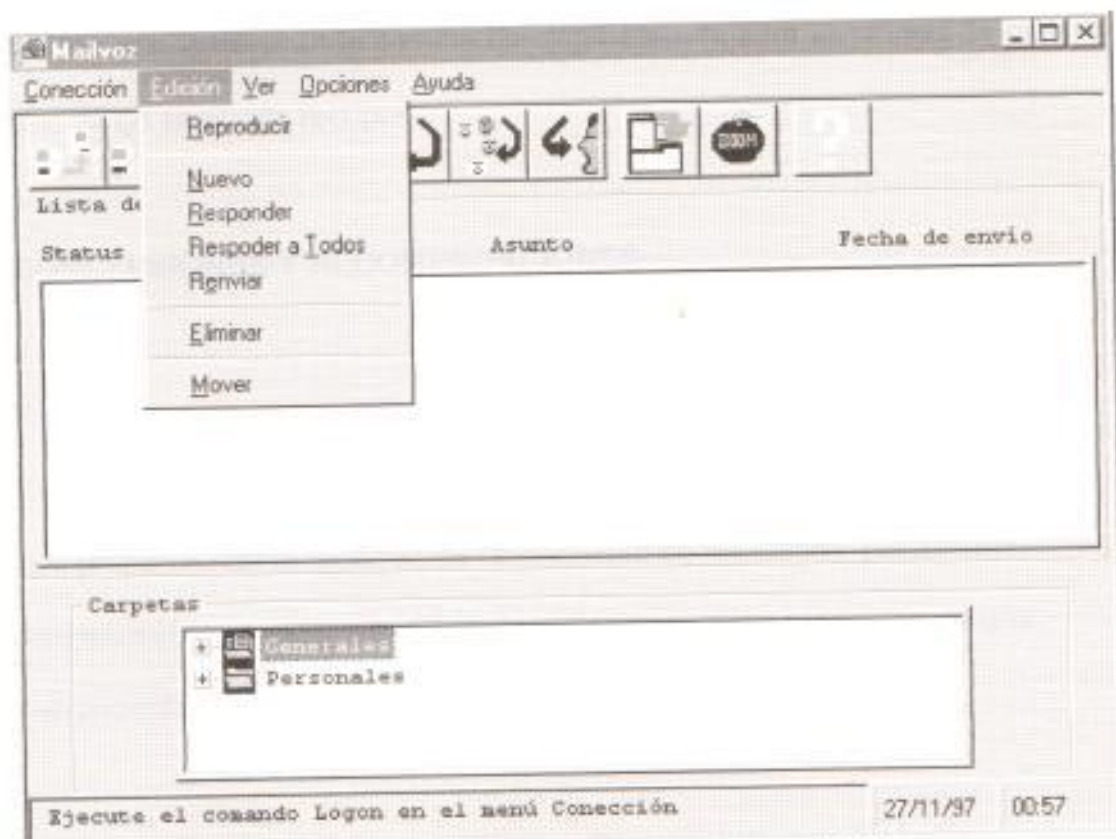


Fig. 71: Pantalla Principal, menú **Edición** se elige la opción **Mover**.

- O se selecciona el siguiente icono:



Fig. 72: Icono para eliminar mensajes

Una vez hecho todos estos pasos el mensaje se elimina pasándolo a la carpeta Basurero que se encuentra en la carpeta Generales.

Un mensaje que se encuentre en la carpeta Basurero puede ser recuperado y leído aún, pero si se lo elimina de dicha carpeta será eliminado de manera que ya no se pueda recuperar.

CONCLUSIONES Y RECOMENDACIONES

Trabajar con un nuevo enfoque, una nueva forma de ver las cosas, otra forma de resolver los problemas que es el análisis orientado a objetos que nos proporcionan la facilidad de reusabilidad de las clases, potabilidad, encriptamiento, encapsulamiento, etc., además de la utilización de una herramienta fuerte y robusta como lo es Visual C++.

La utilización de Messaging Application Program Interface (MAPI) facilitó en mucho el desarrollo del proyecto, ya que el control MAPI Messages efectúa una variedad de funciones de sistemas de mensajería después que se establece una sesión con el control MAPI Session.

Además de utilizar controles existentes como por ejemplo las lista, el grid, etc., nos facilitó la programación.

Para una nueva aplicación se puede recomendar que se usen controles existentes en el mercado que son exclusivos para hacer aplicaciones de

correo, los cuales ya soporta la transmisión de voz con codificación de 64 bits.

Otra forma de poder enviar correo de voz es adjuntando el archivo wav, y poniendo en el cuerpo del mensaje una codificación para diferenciar con los demás mensajes que no sea de este tipo.

Además para un futuro no solo nos limitaremos a enviar nuestra voz sino que también se pueden incluir imágenes, videos, aprovechando de esta manera todas las bondades de los multimedia.

BIBLIOGRAFIA

1. Libro

- a) Microsoft, Inside Visual C++ (4ta. Edición; Microsoft Press, 1996)
- b) Kris Jamsa – Ken Cope, Programación en Internet (1era. Edición; México; McGraw-Hill, 1996), Capítulo 15
- c) Joyanes, Microsoft Visual C++ (1era. Edición; Madrid; McGraw-Hill, 1996),
- d) James Martin – James Odell, Análisis y Diseño Orientado a Objetos (1era. Edición; México; Prentice Hall, 1996)
- e) Microsoft, Visual Basic (1 era. Edición; Indianapolis; Que, 1996), Capítulo 8

2. Reporte técnico

Microsoft Messaging Application Program Interface (MAPI) (Microsoft Corporation, 1993)

APENDICE A

GLOSARIO

Adelantar: Función que adelanta la posición del reproductor en el archivo en tramos indicados por el parámetro DWORD Paso medido en milisegundos.

Adjuntos: Es un objeto CobList que sirve para contener los objetos adjuntos de un mensaje.

AdressRecip: Función que retorna un tipo Cstring con la dirección del recipiente actual.

Anula: Esta función elimina la lista de carpetas, sin parámetros.

AnulaLista: Remueve todos los mensajes de la lista de mensajes.

Atachado: Es una función que recibe un parámetro Cstring que contiene los adjuntos que son sacados del texto leído del archivo por cada mensaje, los separa y los guarda en un mensaje.

AttachmentIndex: Variable tipo POSITION que indica la posición en la lista de recipientes.

Buscar: Verifica si un nombre de carpeta existe, recibe como parámetro un Cstring con el nombre de la carpeta a buscar.

Cabeza: Variable tipo puntero a una estructura tipo LisCar, se la utiliza como la posición inicial de la lista de carpetas.

Cargar: Esta función recibe como parámetro un CString con el nombre del archivo que es donde se va a cargar la lista con los nombres.

CargarMensajes: Esta función abre el archivo tipo texto que contiene los mensajes de una carpeta cuyo nombre es el mismo del archivo, lee el texto de este y usando las funciones encabezado,atachado y receptor forma cada mensaje que luego guarda en la lista de mensajes llamada Mensajes.

CattachmentMsgMsg: Clase que permite representar un adjunto de un mensaje.

CattachmentMsgMsg: Función constructora de CattachmentMsgMsg recibe 2 parámetros name y path que representan el nombre y la ruta del adjunto.

Ccarpeta: Clase que permite almacenar el nombre de las carpetas que existen.

Ccarpeta: Función constructora de la clase del mismo nombre.

ClisMensaje: Clase

Close: Función que cierra cualquier dispositivo con un archivo de audio abierto.

CmapiMessageCtrl: Clase con la propiedad de acceder al servidor y transmitir los mensajes una vez establecida la sesión y de acceder a la información de cada uno de los mensajes que transmite.

CmapiSessionCtrl: Clase cuya misión es establecer la comunicación entre la aplicación cliente y el servidor.

Cmensaje :Función constructora de la clase Cmensaje, que setea el número de recipientes con el parámetro tipo int **numr** y setea el número de adjuntos con el parámetro tipo int **numa**

Cmensaje: Clase que contiene la estructura donde se guarda toda la información de un mensaje y las operaciones para acceder a ella.

Cola: Variable tipo puntero a una estructura tipo LisCar, se la utiliza como la posición final de la lista de carpetas.

Compose: Función que cambia a un objeto de esta clase al modo de escritura de mensajes.

Copy: Función que copia toda la estructura de un mensaje.

CrecipMsg: Función constructora de la clase **CrecipMsg** que setea a **m_Name**, a **m_Adres** y a **m_Tipo** con **CString name, CString adress, short tipo** respectivamente.

CrecipMsg:: Clase que sirve para almacenar los recipientes de un mensaje que está formado por un objeto que pertenece a ésta clase, además posee las propiedades para acceder a un recipiente.

Creproductor: Clase que proporciona los métodos necesarios para poder reproducir un archivo con formato de audio, así como para crear uno nuevo.

Creproductor: Función constructora de la clase **Creproductor** que inicia todos las variables globales.

Delete :Función que borra cualquier elemento de un mensaje, puede ser un recipiente, un adjunto, e incluso el mismo adjunto.

DeleteAttachment: Función que borra un adjunto de la lista de adjuntos y para eso recibe el parámetro **POSITION index** para indicar la posición del adjunto.

DeleteRecip: Función que elimina un recipiente de la lista de Recipientes para ello recibe un parámetro de tipo **POSITION index**.

dwReturn: Variable tipo **DWORD** que recibe la respuesta de error de la función **mciSendCommand**

EliminaElemento: Que recibe como parámetro un Cstring con el nombre de la carpeta que se va a eliminar de la lista de carpetas.

EliminarMensaje: Función booleana que elimina un mensaje de la lista, para ello recibe como parámetro la posición del mensaje dentro de la lista con un tipo int.

Encabezado: Es una función que recibe un parámetro Cstring que contiene el asunto, la fecha, y la bandera no leído que son sacados del texto leído del archivo por cada mensaje, los separa y los guarda en un mensaje.

Fetch: Función que accesa al servidor en busca de nuevos mensajes.

Formato: Variable tipo **DWORD** que sirve para guardar el tipo de formato con que se abre un archivo de audio, medido en milisegundos, minutos, frame, etc.

Forward: Función que copia un mensaje previamente seleccionado con el índice de mensaje al bufer de redacción donde se escriben los nuevos mensajes que van a ser enviados.

GetAttachment: Función que devuelve un puntero a **CAttachmentMsgMsg** con la dirección de un objeto **CAttachmentMsgMsg** dentro de la lista de Adjuntos.

GetDate: Función que devuelve la fecha de creación del mensaje.

GetID: Función que devuelve un **DWORD** con el **ID_Device** que es el identificador del dispositivo abierto para reproducir.

GetNumAttachment: Función que devuelve el número de adjuntos del mensaje.

GetOrigName: Función que devuelve un Cstring con el nombre del remitente de un mensaje.

GetPassword: Función que devuelve un Cstring conteniendo el password del usuario actualmente conectado.

GetRecipIndex: Función que devuelve **RecipIndex** del mensaje como un tipo **POSITION**.

GetRecipMsg: Función que devuelve un puntero a **CrecipMsg** con la dirección de un objeto **CrecipMsg** dentro de la lista de Recipientes.

GetSessionID: Función que devuelve un long con el ID de la sesión establecida.

GetStatus: Función que consigue los datos que definen el estado de un dispositivo tales como la longitud, el formato etc., dependiendo de las banderas que se envíen como parámetros en **DWORD Flag**.

GetSubject: Función que devuelve la fecha de creación del mensaje.

GetUnReady: Función que devuelve el estado de no leído del mensaje como un booleano.

GetUserName: Función que devuelve un Cstring conteniendo el nombre del usuario actualmente conectado.

Guardar: Función que guarda el archivo de audio creado por **Rec** con la ruta completa ingresada en el parámetro **const char* file**.

Guardar: Esta función recibe como parámetro un CString con el nombre del archivo que es donde se va a guardar la lista con los nombres.

GuardarAdjuntos: Es una función que recibe un parámetro Cstring donde se almacenarán los adjuntos de un mensaje guardado en una variable temporal tipo mensaje.

GuardarEncabezado: Es una función que recibe un parámetro Cstring donde se almacenará el encabezado de un mensaje guardado en una variable temporal tipo mensaje.

GuardarMensajes: Esta función abre el archivo tipo texto que contendrá los mensajes de una carpeta cuyo nombre es el mismo del archivo, carga al texto mensaje por mensaje usando las funciones GuardarEncabezado, GuardarAdjuntos, GuardarRecipientes, formando un sólo texto con todos los mensajes, luego guarda en el archivo llamado con el nombre de la carpeta cuya lista de mensajes se desea guardar.

GuardarRecipientes: Es una función que recibe un parámetro Cstring donde se almacenarán los recipientes de un mensaje guardado en una variable temporal tipo mensaje.

InsertAttachment Función que recibe los siguientes parámetros **CString path, CString name** para setear la ruta y el nombre del adjunto respectivamente para insertar uno en la lista de Adjuntos.

InsertRecip: Función que recibe los siguientes parámetros **CString name, CString adress, short tipo**, para setear el nombre, la dirección y el tipo de recipiente respectivamente de un recipiente nuevo que se va a insertar a la lista de Recipientes.

int GetNumRecip: Función que devuelve el número de recipientes del mensaje.

Longitud: Variable tipo **DWORD** que sirve para recibir la longitud de un archivo de audio.

m_Adress: Variable tipo Cstring que contiene la dirección de un recipiente.

m_Date: Variable tipo Cstring que contiene la fecha de creación del mensaje.

m_Name: Variable tipo Cstring que contiene el nombre de un recipiente.

m_NameFile: Variable tipo CString para almacenar el nombre del adjunto.

m_NamePathFile: Variable tipo CString para almacenar la ruta del adjunto.

m_NumAdj: Variable tipo int que contiene el número de los adjuntos de un mensaje.

m_NumRecip: Variable tipo int que contiene el número de los recipientes de un mensaje.

m_Subject: Variable tipo Cstring que guarda el asunto del mensaje.

m_Tipo: Variable tipo int que contiene el tipo de un recipiente que es: 0 para el remitente, 1 para los destinatarios originales y 2 para los destinatarios con copia

m_UnReady: Variable booleana que guarda el estado de no leído.

Mensajes: Es un objeto tipo CobList cuya función es almacenar los mensajes.

MensajesCont: Devuelve la cantidad de mensajes que tiene almacenados la lista Mensajes.

MeteElemento: Función que recibe como parámetro un CString con el nombre de la carpeta que se va a ingresar a la lista de carpetas y lo inserta en la posición que le corresponde en orden alfabético ascendente.

NameAttachment: Función retorna un CString que es el nombre del adjunto.

NameRecip: Función que retorna un tipo CString con el nombre del recipiente actual.

Next: Función que setea al final la posición del reproductor en el archivo.

OpenFile: Función que abre el dispositivo con el archivo de audio indicados en los parámetros char* file, char* Dispositivo respectivamente.

PathFile: Función retorna un CString que es la ruta del adjunto.

PauRes : Variable tipo **BOOL** que sirve de bandera para saber si el reproductor tiene el dispositivo pausado.

Pause: Función que pausa el dispositivo del reproductor y devuelve un booleano indicando si lo hizo o no.

Play: Función que inicia la reproducción del archivo de audio por parte del dispositivo del Reproductor.

Posicion: Variable posición **DWORD** que recibe la posición en que el reproductor se ha fijado en el archivo de audio.

Posicion: Variable tipo puntero a una estructura tipo LisCar, se la utiliza como la posición actual de la cual se quiere extraer el nombre de la carpeta.

Prev: Función que setea al inicio la posición del reproductor en el archivo.

Rec: Función que usa el dispositivo para grabar desde la posición que se encuentre en el archivo de audio abierto en caso de haberlo sino crea uno y empieza a grabar desde el principio.

Receptor: Es una función que recibe un parámetro Cstring que contiene los recipientes que son sacados del texto leído del archivo por cada mensaje , los separa y los guarda en un mensaje.

Recipientes: Es un objeto CobList que sirve para contener los objetos recipientes de un mensaje.

RecipIndex: Variable tipo POSITION que indica la posición en la lista de recipientes.

Reply: Función que copia al bufer de redacción un mensaje nuevo con la dirección del remitente de un mensaje seleccionado con el índice de mensaje como destinatario .

ReplyAll: Función que copia al bufer de redacción un mensaje nuevo con la dirección del remitente de un mensaje seleccionado con el índice de mensaje como destinatario y los demás destinatarios originales como con copia .

ResolveName: Función que comprueba si un nombre aparece en la lista de direcciones o no y si no está manda un mensaje de error.

Retroceder: Función que retrocede la posición del reproductor en el archivo en tramos indicados por el parámetro **DWORD Paso** medido en milisegundos.

SacaElemento: Función que retorna el nombre de la carpeta contenida en la posición actual , que es de tipo CString no lleva parámetros.

Save: Guarda todo el mensaje recibido previamente seleccionado en un archivo.

Send: Función enviar transmite un mensaje previamente escrito en el bufer de redacción y seteado sus destinatarios, siempre y cuando exista una sesión válida.

SetAttachmentIndex: Función que setea la variable **AttachmentIndex** de un mensaje y recibe como parámetro un tipo **POSITION index**.

SetDate: Función que setea la fecha de creación del mensaje y para eso recibe un parámetro tipo **CString date** .

SetMsgIndex: Devuelve el puntero al mensaje de la lista, indexado con la posición que recibe como parámetro que es de tipo int.

SetPassword: Función que setea el nombre del usuario

SetRecipIndex: Función que setea la variable **RecipIndex** de un mensaje y recibe como parámetro un tipo **POSITION index**.

SetSubject: Función que setea el asunto de un mensaje y recibe como parámetro un tipo **CString subject**

SetUnReady: Función que setea la bandera de no leído de un mensaje y recibe como parámetro un tipo booleano.

SetUp: Función que permite posicionarse al inicio de la lista de carpetas.

SetUserName: Función que setea el nombre del usuario.

Show: Función que muestra la libreta de direcciones, previamente seteada con un parámetro variante para setear el modo como la va a mostrar.

SignOff: Función que finaliza la sesión y se desconecta del servidor.

SignOn: Función que realiza la conexión con el servidor y consigue el ID de la sesión de parte del servidor.

Status: Variable que contiene el estado en que se encuentra el dispositivo del Reproductor.

Stop: Función que detiene la reproducción de un archivo.

Temporal: Variable tipo puntero Cmensaje que sirve para obtener la estructura de un mensaje con el fin de acceder a su información.

TipeRecip: Función que retorna un tipo int con el tipo del recipiente actual.

Vacia: Función que retorna un BOOL para saber si la lista está cargada o vacía, sin parámetros.

wDeviceID : Variable tipo UINT que recibe el identificador del dispositivo de audio abierto.