



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

ARAGON

7
2ej.

Paginación

**"SISTEMA INTEGRAL DE INDEMNIZACION A
SINIESTROS AUTOS DE SEGUROS MONTERREY AETNA
(MODULO DE RADIO)"**

T E S I S

QUE PARA OBTENER EL TITULO DE

"INGENIERO EN COMPUTACION"

PRESENTA:

VERONICA BARRERA FLORES

ASESOR:

ING. ERNESTO PEÑALOZA ROMERO

*SAN JUAN DE ARAGON, EDO. DE MEXICO
AGOSTO DE 1998*

**TESIS CON
FALLA DE ORIGEN**

265350



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mis padres

Santiago Barrera y María Verónica Flores,
por su apoyo, comprensión y paciencia en
la finalización de mi profesión.

A mis hermanos

Jorge y Azael, por el apoyo y el
entusiasmo brindado para concluir
mi tarea.

A mis profesores

Por compartir conmigo sus infinitos
conocimientos

INDICE

Introducción	3
Capitulo I. Análisis del SIISA	6
1.1 Introducción al SIISA	7
1.1.1 Funcionamiento del Sistema	12
1.2 Miniespecificaciones del Sistema	17
1.3 Requerimientos del Sistema	23
1.4 Diagramas de Transición de Estados	24
Capitulo II. Desarrollo del Sistema	27
2.1 Prototipos.	28
2.2 Componentes	34
2.2.1 Tipos de Componentes	35
2.2.2 Servicio del Componente	36
2.2.3 Ambito del Componente	38
2.2.4 Descripción del Componente	38
2.2.5 Clases del Componente	38
2.2.6 Formas	41
2.2.7 Datos Generales	41
2.2.8 Componentes que se integraron al Módulo de Radio	41
Capitulo III Implementación	43
3.1 Base de Datos para el Desarrollo del Modulo de Radio	44
3.2 Programación de Prototipos	47
3.3 Instalación	58

INTRODUCCION

Dentro del crecimiento y desarrollo tan rápido que ha tenido la ciencia de la computación, sobre todo en el surgimiento de nuevos lenguajes de programación, se ha tenido la necesidad de cambiar los programas viejos creados en lenguajes hechos en ambiente de MS-DOS a un ambiente gráfico de WINDOWS 95, por ejemplo Visual Basic, Visual C, etc.

Al tener nuevas herramientas para poder desarrollar otro tipo de programas muchas empresas han tenido la necesidad de cambiar muchos programas para poder tener un mejor servicio.

Un ejemplo lo podemos encontrar en uno de los mercados más grandes que tenemos no nada más en nuestro país, que es la venta de autos, pero las agencias al vender un auto también venden lo que es un seguro de auto, esto es con el fin de tener alguna garantía en caso de cualquier accidente o robo.

Confirmando con las estadísticas, la venta de autos a incrementado, por lo tanto muchas de las empresas de seguros de autos debe competir dando un mejor servicio a sus clientes.

Para este caso Seguros Monterrey Aetna cuenta con un sistema de atención a cualquier tipo de siniestros¹ autos, el cual es muy lento ya que es manejado dentro de un ambiente de MS-DOS y no es muy flexible, por que no tiene acceso a un nivel nacional y como en la actualidad han crecido sus clientes se han visto en la necesidad de mejorar sus sistemas a un ambiente gráfico y flexible.

Lo más importantes para cualquier Aseguradora es la atención al cliente, y el sistema contiene un módulo llamado Radio, en el cual se registran los tiempos de cada ajustador². Por tal motivo este proyecto se dedicara a este módulo.

Es importante hacer un análisis del sistema actual para poder conocer el funcionamiento y manejo y así encontrar los posibles requerimientos, cambios en aquellas partes que no se utilicen y así poder contar con un proyecto mejor, toda esta información se podrá encontrar en el Capitulo I.

Al iniciar el desarrollo del sistema se requieren tener prototipos lo cual es la base para poder iniciar un proyecto y poder darle el flujo de datos para que posteriormente se pueda finalizar, de esto hablaremos en el Capitulo II

¹ Siniestro.- Daño o desgracia que sufren las personas o la propiedad, por muerte, accidente, incendio, etc.

² Ajustador.- Persona que se encarga de atender un siniestro en crucero. Estos pueden ser de dos tipos: ajustador de auto y ajustador de moto

Al terminar el desarrollo culminaremos con la parte más interesante que es la implementación de nuestro proyecto, esto es, para que pueda ser manejado por todos los usuarios que atienden este tipo de siniestros, el cual se explicará en el Capítulo III.

De esta forma como Ingenieros en Computación podemos ayudar en el mejoramiento de un sistema para las necesidades de nuestra sociedad.

CAPITULO I

ANALISIS DEL SIISA

Para poder hacer un buen desarrollo para todo tipo de sistema, necesitamos conocer las características, especificaciones, requerimientos que lo componen, esto es, para poder hacer un análisis y posteriormente desarrollarlo con mayor facilidad.

Este capítulo nos introducirá al conocimiento del sistema y sus características generales, pero nos enfocaremos más ampliamente en el análisis del módulo de Radio.

1.1. INTRODUCCION AL SIISA

El Sistema Integral de Indemnizaciones a Siniestros Autos de Seguros Monterrey Aetna se encarga de dar atención a personas que tuvieron algún accidente o siniestro como por ejemplo: robo de auto, colisión, atropellamiento, etc.

Seguros Monterrey Aetna cuenta con seis divisiones, estas se distribuyen la carga de trabajo por los diferentes estados de la República Mexicana, los cuales se muestran en la figura 1.

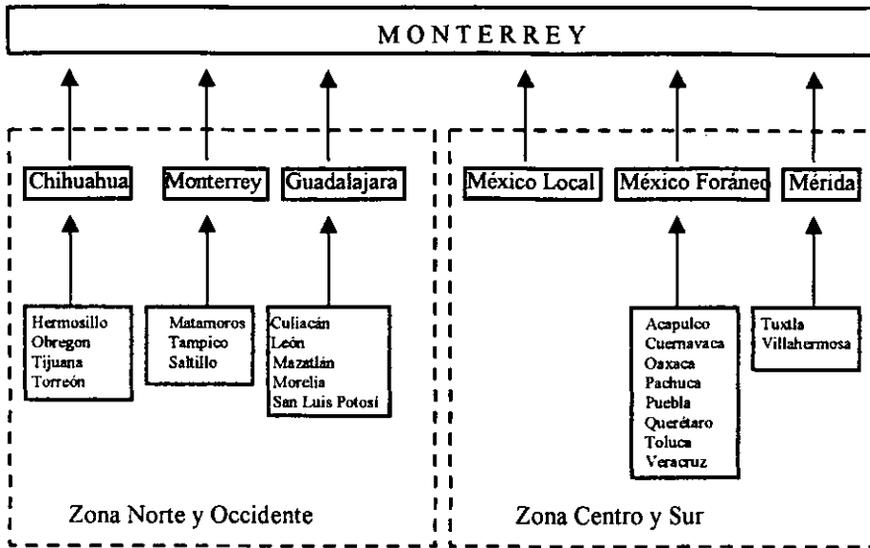


Figura 1

El sistema SIISA actual, como ya se dijo en la introducción es un sistema con ambiente de MS-DOS bajo un lenguaje de programación de FOX-PRO, el cual una de las desventajas más grandes que contiene este lenguaje es que es muy difícil transmitir los datos a través de red y su proceso es muy lento.

EL programa SIISA es muy grande el cual contiene los siguientes elementos para poder operar:

Tablas	400
Programas	965 Programas 262 Rutinas
Módulos	18 Módulos
Reportes	415
Interfaces	38 Procesos 61 Archivos

Lo mostrado anteriormente estaba operando, por lo que se puede observar son cantidades muy grandes, ya que para poder verificar el funcionamiento de 965 programas y 262 rutinas es demasiado lento, y de los cuales 75 programas y 36 rutinas son las que utiliza el módulo de Radio.

El número de oficinas donde se opera el sistema SIISA se muestra en la siguiente tabla:

Oficinas	Estaciones
Monterrey	68
Lago Xochimilco (México D. F.)	61
Calzada del Hueso (Méx. D. F.)	22
Guadalajara	40
Chihuahua	11
Mérida	14
Total de estaciones	216

Cada una de estas oficinas cuentan con una red interna las cuales hacían operar el sistema, para varios ordenadores.

Este sistema cuenta con 18 módulos los cuales están divididos en dos tipos los cuales se muestra en la siguiente tabla

Módulos Operacionales	Módulos de Apoyo
Atención	Actualizadores
Radio	Integra
Talleres	Servicios externos
Ajustes	Reportes
Valuación	
Proveedores	
Indemnizaciones	
Cajas	
Recuperación de Robos	
Salvamento	
Históricos	
Emisión y Cobranza	
Recepción	
Cobros	

La razón de efectuar una conversión del sistema SIISA a un sistema más rápido y seguro es por las numerosas tablas que se estaban utilizando los requerimientos técnicos y generales son los siguientes:

- *Características Generales*

El sistema contiene información en producción e histórica, las cuales se requieren trasladar al nuevo sistema al momento de implementar cada una de las oficinas.

Las fases consideradas para la conversión comprenden, tanto para el funcionamiento actual como para el funcionamiento de nuevos requerimientos

- *Características Técnicas:*

Aplicará el proceso de normalización al nuevo modelo de la Base de Datos (BDD), ya que se hará el cambio de Fox a Informix

Se reducirá la cantidad de índices declarados en la BDD, debido principalmente a la reducción de reportes de control de procesos (en línea).

Que la aplicación quede preparada para centralizarla para todas las oficinas que conforma SMA.

Por lo tanto la conversión del sistema quedara de la siguiente manera

	Actual	Conversión
Sistema Operativo	Dos y Novell 4.10	Unix HP-UX 9.0
Base de Datos	Fox-Pro 2.5	Informix 7.12
Front End	Fox-Pro	Visual Basic 5.0
Protocolo de comunicaciones	IPX	TCP/IP
Conectividad		ODBC I-CLI 2.5 WC3
Reporteador		Impromptu

1.1.1. Funcionamiento del Sistema

El funcionamiento del sistema comienza desde que se registra una llamada telefónica en el módulo de *Cabina*, la persona que atiende la llamada (Cabinero) registra su número de póliza y su nombre para poder tener todos sus datos y posteriormente confirmarlos, se pueden tener varios tipos de siniestros los cuales son la colisión o choque, el atropellamiento, el robo, o daños provocados por terceros desconocidos, ya sea en un lugar fijo o en movimiento.

Para todos los casos de siniestros, el cabinero tiene la obligación de asignarle un número de siniestro y dependiendo el tipo de siniestro se continua con el siguiente módulo, por ejemplo si es colisión, atropellamiento o daños provocados por terceros

desconocidos se continua la atención en el siguiente módulo (Radio), si es un robo se toman sus datos particulares para que un ajustador pueda visitarlo posteriormente y atender su siniestro personalmente.

Radio.

En el módulo de Radio es uno de los procesos importantes para la atención del asegurado ya que entre más rápido sea atendido, se obtendrán mejores resultado.

El proceso comienza cuando se registra un siniestro en una pantalla de listado amplio, cuando la persona encargada de este módulo se notifica que hay un nuevo siniestro, este asigna un ajustador de auto para que vaya al lugar del siniestro y atiende personalmente el problema que se reporto, la manera en como se comunica con el ajustador es por Radio C. B. y por medio de mensajes en Skaytel le manda la información sobre el siniestro que se le asigno (número de siniestro, lugar donde se efectuó, tipo de auto, e información sobre la póliza), posteriormente hace un anuncio por radio para que los ajustadores de moto más cercano al lugar del siniestro mencionado se puedan trasladar más rápidamente y poderlo atender mientras el ajustador de auto llega y continua atendiendo el siniestro y se libera el ajustador de moto para que este listo a otro llamado.

Esta persona también se encarga de registrar los tiempos de aviso a cada ajustador, tiempo de llegada, tiempo de retiro del asegurado y tiempo de termino de la atención.

Un ajustador de auto o de moto se puede liberar por diferentes razones que son las siguientes:

- No le es posible llegar al ajustador al lugar del siniestro.
- Existe otro ajustador más cerca del lugar del siniestro.
- El ajustador no encontró los vehículos en el lugar.
- El siniestro va a ser turnado a alguna dependencia. Esta opción deja el siniestro en estatus de pendiente y listo para asignarle un ajustador de dependencia.
- El asegurado llegó a un acuerdo con el tercero
- El asegurado no tiene póliza vigente con la aseguradora
- El siniestro no procede por cobranza de la póliza
- El siniestro no procede por vigencia de la póliza

Se puede cancelar el siniestro siempre y cuando se haya asignado ajustador al mismo. Las causas de cancelación son las siguientes:

- No fue posible brindar la atención en el crucero
- El asegurado no requirió atención en el crucero

También cuando se tiene una visita de un caso de robo, el día en que se le dio la cita se registra también en el listado amplio para poder atender a la persona.

Otra tarea muy importante en este módulo es reactivar siniestro cancelados por no localizados, también podemos corregir los tiempos de atención de siniestros en cruceo y desde luego se encarga de asignar ajustador a los siniestros que fueron determinados como pendientes, en otras palabras, se le asigna un ajustador de dependencia.

Podemos hacer consultas de la Cobranza de pólizas y podremos ver en detalle los recibos pagados que se encuentran registrados en el sistema.

Al finalizar la atención dependiendo del tipo de siniestros pasan a la parte de *Declaración* del siniestro el cual los ajustadores registran los datos tomados en la parte de atención y es donde se declara realmente el tipo de siniestro.

En el proceso de atención dependiendo el grado en que se encuentre el vehículo, este podrá solicitar un pase para un taller y poder ser reparado con rapidez, si el vehículo por algún motivo no puede circular, este también puede solicitar una grúa al módulo de Talleres, el cual la petición se realiza por radio y la confirmación por radio y vía Skaytel.

Posteriormente se continuo con la parte de valuación, en este módulo se registra los daños que encontró un valuator, los cuales se podrán confirmar en un vídeo que se tomo al momento de ser valuado.

Al registrar todos los daños en el sistema, estos se registran en un sistema llamado VALUADATA³ el cual se encarga de valorar en costos los daños del vehículo y da posibles proveedores de refacciones.

En el caso de un robo, este se registra en el módulo de *Recuperación de Robos* y se le notifica a AMIS, enviando los datos por otro programa externo al SIISA, el cual se encarga de ayudar en la búsqueda del vehículo. Si el vehículo es encontrado en un lapso no menor a 28 días, este se le comunica al asegurado y si tiene alguna avería se arregla y se hace entrega, pero si rebasa el límite señalado, pasa a ser propiedad de la aseguradora y se liquida al asegurado la parte acordada del costo del vehículo; en este caso cuando el auto se encontró con pequeños defectos, este se utiliza para trabajo de la empresa y en el caso de tener mayores daños, si se pueden reparar se efectúa un salvamento (y un número de salvamento para un control), y se registra en el módulo con el mismo nombre, el cual lleva a cabo una subasta internamente de la aseguradora y de ahí recupera una parte de lo que se invirtió.

También encontramos el módulo de *Indemnizaciones*, este se encarga de los afectados y terceros que tuvieron algún daño dentro de la colisión y este evalúa los daños e los indemniza.

El módulo de *Cajas* se encarga de registrar y realizar los cheques que registra el módulo de indemnizaciones, valuación, talleres, salvamentos.

³ Valuadata.- Sistema externo que realiza cálculos de costos para daños de un vehículo

Faltan de mencionar varios módulos, pero estos se encargan de llevar la contabilidad y los siniestros históricos de hace 5 años los cuales también forman parte importante para poder atender un solo siniestro.

1.2. MINIESPECIFICACIONES DEL MODULO DE RADIO

Las miniespecificaciones que se obtuvieron de los procesos del módulo de Radio son los siguientes:

<i>Proceso</i> : Controlador de Radio
<i>Flujo de datos de Entrada</i> : Ajustador, Poliza, Siniestro, citas, horario, horanoct, hora_las, citas
<i>Flujo de datos de Salida</i> : Ajustador, Siniestro, aux-ajus, autoaseg, sin-rod, citas, horanoct, horario, zona, hora-las
<i>Datos de entrada almacenados:</i> Datos del Vehículo, Datos del Asegurado Datos del lugar, Fecha y hora del siniestro
<i>Datos de salida almacenados :</i> Horarios de los ajustadores, estatus del siniestro

Términos locales:

- Solo se atienden siniestros en crucero
- Primero se asigna el ajustador de auto
- Cuando no hay libre un ajustador de auto se asigna uno de moto
- Al actualizar siniestros pendientes, esto significa que no se conoce la compañía aseguradora (ajustador XX)

Función

- Controla los tiempos que corresponde a cada ajustador por siniestro.
- Si se cancela un siniestro sólo cambia el estatus del siniestro dependiendo del motivo de la cancelación
- Si se asigna un ajustador el estatus del siniestro cambio a 01
- Si se libera un ajustador, en la tabla de ajustador y aux_aius reemplaza por blanco todos los datos capturados anteriormente y su estatus es 00
- Al finalizar de atender un siniestro este tendrá un estatus 01

<i>Proceso</i> : Mensaje de Skaytel a los ajustadores
<i>Flujo de datos de Entrada</i> : Siniestro, cat-aseg
<i>Flujo de datos de Salida</i> : Siniestro, cat-aseg
<i>Datos de entrada almacenados:</i> Datos del asegurado Datos del vehículo Estatus de la Póliza Datos del lugar del Siniestro
<i>Datos de salida almacenados:</i> Datos del asegurado Datos del vehículo Estatus de la Póliza Datos del lugar del Siniestro

Términos locales

- Se toman los datos del siniestro, del asegurado y del vehículo

Función

- Manda por Skaytel el mensaje a cada ajustado (Auto o Moto)

<i>Proceso : Liberar Ajustador</i>
<i>Flujo de datos de Entrada: Motivos de la liberación</i>
<i>Flujo de datos de Salida : Motivo de la liberación, aux-ajus, ajustado</i>
<i>Datos de entrada almacenados:</i> Ajustado, aux-ajus, zona
<i>Datos de salida almacenados :</i> ajustado, aux-ajus, zona

Términos locales

Dependiendo de la causa de la liberación va a realizar diferente actividad, por ejemplo:

- En la causa 1 y 2: solo pide la zona y reemplaza con blancos todos los datos del ajustador y las horas de las tablas mencionadas
- En la causa 3: se reemplaza el status a 91 y reemplaza con datos correspondientes al ajustador, zona y tiempos
- EN causa 4: pide la zona para asignarle la dependencia y reemplaza con blancos los datos al ajustador, zona y tiempos
- En causa 5: reemplazo el status a 92 y reemplaza con blancos los datos del ajustador, zona y tiempos
- En causa 6: reemplaza el status a 90 y reemplaza con datos correspondientes al ajustador, zona y tiempos
- En causa 7: reemplaza el status a 93 y reemplaza con datos correspondientes al ajustador, zona y tiempos
- En causa 8: reemplaza el status a 94 y reemplaza con datos correspondientes al

- ajustador, zona y tiempos
- Para todos los casos pasar la información a insunix

Función

- Cancela o libera el ajustador de un siniestro

<i>Proceso:</i> Corrección de Tiempos
<i>Flujo de datos de Entrada :</i> Siniestro, aux-ajus, ajustador
<i>Flujo de datos de Salida :</i> siniestro, aux_ajus, ajustador
<i>Datos de entrada almacenados:</i> No. Siniestro Datos ajustador Tiempos de atención
<i>Datos de salida almacenados:</i> No. Siniestro Datos ajustador Tiempos de atención

Términos locales

- Se podrá remplazar los datos del ajustador y los tiempos de atención

Función

- Cambiar los tiempos de atención y el ajustador que lo otendio.

<i>Proceso:</i> Asignación de ajustador de dependencia
<i>Flujo de datos de Entrada :</i> siniestro, poliza, ajustador, aux-ajus, zona, autoaseg, hora-las
<i>Flujo de datos de Salida :</i> siniestro, aux-ajus, autoaseg
<i>Datos de entrada almacenados:</i> Datos del siniestro Datos del ajustador Datos del asegurado
<i>Datos de salida almacenados:</i> Ajustador (asignado) Siniestro Aux-ajus

Términos locales

- Se anexara en aux-ajus, siniestro, autoaseg, el siniestro con todos los datos si ya se encuentra solo reemplaza o anexo la información faltante

Función

- Asigna el ajustador de la dependencia

<i>Proceso</i> : Actualiza siniestro pendientes
<i>Flujo de datos de Entrada</i> : siniestro, poliza, ajustador
<i>Flujo de datos de Salida</i> : siniestro
<i>Datos de entrada almacenados:</i> Siniestro Poliza
<i>Datos de salida almacenados:</i> Siniestro

Términos locales

- Se actualizara el número de siniestro

Función

- Poder cambiar de analista y número de siniestro

1.3. REQUERIMIENTOS DEL SISTEMA

Para empezar el desarrollo del sistema se tuvieron que revisar los requerimientos para el nuevo sistema.

- Tener en el listado amplio la asignación del ajustador en una pantalla, esto es para ahorrar tiempo, ya que hay ocasiones que es bastante la cantidad de siniestros que no se pueden manejar tan fácilmente.

- Se tuvieran en la misma pantalla (listado amplio), la forma de enviar mensajes a lo ajustadores, de tal forma que sea más fácil el manejo del módulo y
- Por último que fuera más rápido el acceso y el flujo de información.

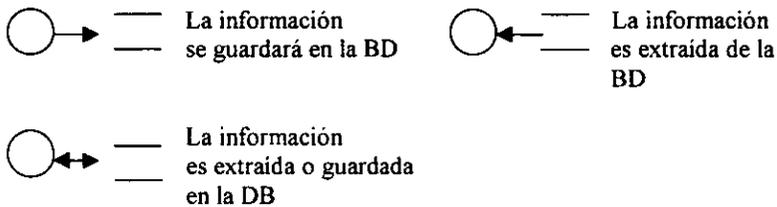
1.4. DIAGRAMAS DE TRANSICIÓN DE ESTADOS

Un diagrama de transición de estados son aquellos que por medio de gráficas de Gantt podemos mostrar los diferentes procesos y bases de datos que cuenta el sistema

Nosotros podemos representar un proceso por medio de un círculo, en donde el cual se representa una acción importante del sistema

Las dos líneas paralelas representan una tabla o comúnmente llamada Base de Datos (DB), en donde de van a almacenar o leer los datos respectivos al proceso que se este manejando

Las líneas representan el flujo de información, por ejemplo



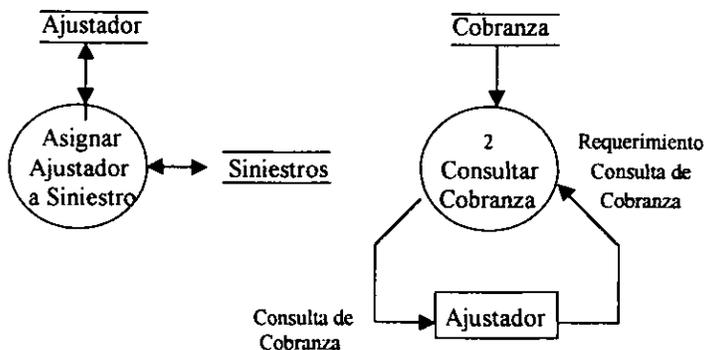
Por medio de niveles podemos representar nuestro sistema, empezando por el nivel CERO, el cual representa el sistema en general, posteriormente se van desarrollando los siguientes niveles conforme los procesos que se vayan generando del general.

Es así como se desarrolla la gráfica de estados del Módulo de Radio

Nivel Cero

El nivel cero se puede dividir en dos secciones, las cuales son importantes para el desarrollo de módulo, en la primera parte se tiene como acción la asignación del ajustador, en donde se tiene como tabla a Ajustador en donde vamos a encontrar todos

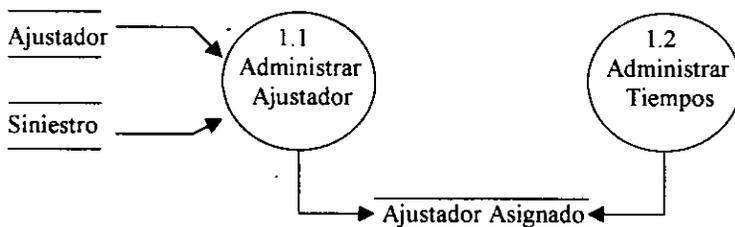
los ajustadores habilitados y se asignaran a aquellos siniestros que se encuentran en proceso de atención. Como se muestra en la siguiente figura.



En la segunda parte del nivel cero se tiene una consulta de la cobranza el cual se va a tener como tabla Cobranza y esta información la podrá obtener el ajustador, el cual verifica si es vigente la cobranza. Como se mostró en la figura anterior.

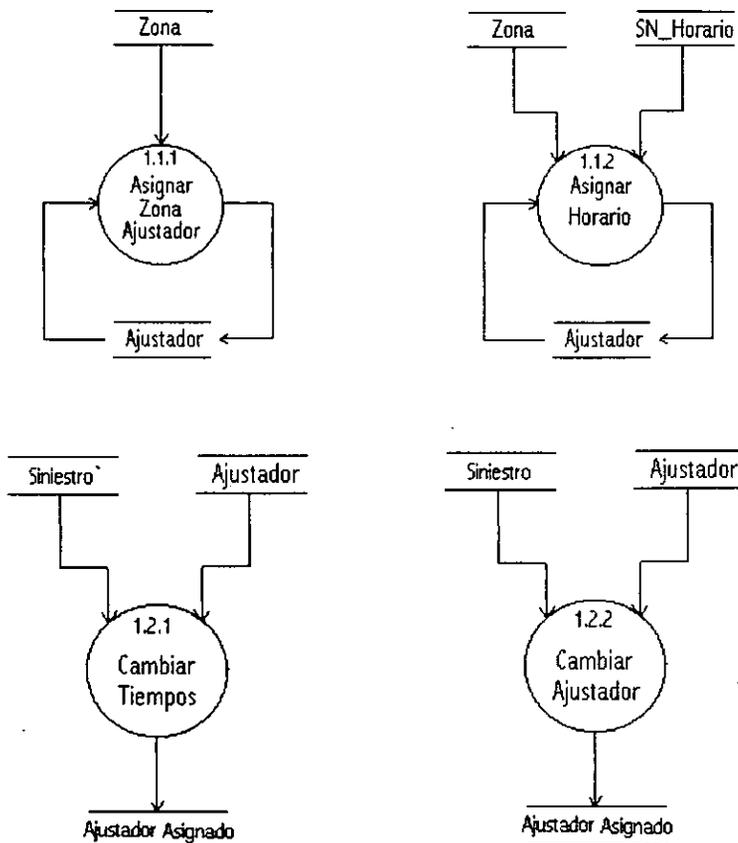
Nivel Uno

El nivel uno es la parte donde se desarrollan más tareas que componen el nivel cero, en este caso, dentro de la asignación de ajustadores, podemos administrar tanto a los ajustadores como sus tiempos de aviso, arribo, retiro del asegurado y terminación de la atención, en la cual se guarda todos estos datos en la tabla llamada Ajustador Asignado. Se representa en la siguiente figura:



Nivel Dos

Si se tiene mas tareas dentro de las tareas realizadas anteriormente podemos desglosarla en otro nivel mas, en esta caso tenemos mas tareas que realiza cada una de las anteriores acciones las cuales se dividen en la Asignación de Zona a un ajustador, la asignación de un horario esta parte corresponde a la administración del ajustador y con respecto a la administración de tiempos podemos encontrar el cambio de tiempos y el cambio de ajustadores los cuales podremos representar con las siguientes figuras:



CAPITULO II

DESARROLLO DEL SISTEMA

Después de haber analizado el sistema, obteniendo sus especificaciones, y haber realizado los diagramas correspondientes podremos diseñar los prototipos y verificar los componentes, los cuales nos servirán de referencia para poder desarrollar la programación posteriormente.

2.1 PROTOTIPOS

Los siguientes prototipos fueron diseñados conforme al anterior sistema SIISA, el cual es diferente ya que contiene los nuevos requerimientos que el usuario necesita, para un mejor manejo y ambiente.

Atención a Siniestros

Atención de Siniestros se divide en dos partes en Listado amplio y en Reactivación de siniestros, en donde se requiere la atención rápida y eficaz, tanto de los ajustadores como del Radio-Operador.

BASE DE DATOS									
Control de Radio		Corrección de Tiempos		Asignación Ajustadores Dependencia		Estrategia de 256.000		Consulta de Cambios	
Listado Amplio		Reactivar Siniestros		Lugar del Siniestro		A/A		Atendidos	
Nº	CITA	18172	RIO SAN JOAQUIN						
1	CITA	13470	AV CENTRAL ESQ AV 412 COL SAN JUAN DE ARAOON						
2		9076	AV. RIO SAN JOAQUIN						
3		13572	JUNTO AL MONUMENTO A CUALTEMOC						
4		14175	PRESIDENTE MASARYK Y MARINA NACIONAL						
5		15878	INSURGENTES SUR #48						
6	CITA	18871	MEXICO TACUBA	A3A	X70	28/12/1997	17:55:00	26418430	1
7		17980	INSURGENTES SUR #45 Y MIGUEL ANGEL DE QUEVEDO	A2B	X67	03/01/1998	16:40:00	9654830	1
8		18089	INSURGENTES SUR #45 Y MIGUEL ANGEL DE QUEVEDO			03/01/1998	16:40:00	5421230	1
9		19689	ASDA	A2C	X76	13/01/1998	21:51:00	4211630	1
10		19884	SAFO			13/01/1998	22:55:00	4125230	1
11	CITA	19987				13/01/1998	22:57:00	3520430	1
12		20084	SDFMSDFGUSDFLQKJ	R13		13/01/1998	23:06:00	2138030	1
13		20280	REFORMA	R14		13/01/1998	23:34:00	1619630	1
14		21283	DFGDSGFSGDS	A1C	X26	14/01/1998	18:14:00	755630	1
15		21489	REFORMA # 485			14/01/1998	21:46:00	668230	1
16								668230	1
17									

- *Listado Amplio.*

En el listado amplio se pueden observar que se realizan varias operaciones, el cual el radiooperador tendrá que ser ágil para dar un servicio adecuado a todos los siniestros que se encuentran en atención (Estos se pueden observar en el grid). Al elegir un siniestro del listado se podrán asignar tanto ajustadores de Auto como de Moto y asignarles tiempos de aviso, llegada, retiro del asegurado y termino, a cada uno de ellos, también se podrá enviar mensajes a los correspondientes ajustadores, consultar las citas si es que tiene el siniestro a atender, consultar los datos más importantes del siniestro por medio de la consulta por F2 o por los binoculares, una parte muy importante es la liberación del ajustador o la cancelación del siniestro esto se muestra con los botones que se encuentran, y también podrá consultar la cobranza de la póliza que esta en atención.

Reactivación de Siniestros Cancelados

En esta parte se reactivan a aquellos siniestros que se cancelaron porque el ajustador no encontró a los automóviles en el lugar donde se registro el siniestro, es decir, si en caso de que el ajustador los encontrara posteriormente se reactiva y se le da el estatus de atención y este aparecerá de nuevo en el listado amplio para asignarles nuevamente ajustador y los tiempos de seguimiento del siniestro.

Controlador de Platos | Cobertura de Tránsito | Asignación de Ajustadores | Desempeño | Estatus de Cambios | Siniestros

Listado Amplio | **Reactivar Siniestros**

Descripción del Siniestro

Siniestro: [] Zona: [13 Del Alvaro Obregón]

Póliza: [443925] [3003] Asegurado: [T. Susana M. Torres Andra]

Fecha Siniestro: [30/04/98] [17:45:36] Fecha de Revisión: [30/04/98] [17:45:36]

Lugar de Siniestro: [Cada del Hueso con Av. de la Paz]

Descripción del Auto

Modelo: [1997] Placa: [L-C4-B]

Marca: [Daimler]

Reactivar Siniestro

Reactivación de Siniestros | LAPS | NUM | INS | 01:37 | 09/04/1998

Corrección de tiempos

Aquí pueden corregirse los tiempos de atención a los siniestros que ya no están en atención, pueden corregirse los tiempos siempre y cuando sean de acuerdo a la hora del reporte del siniestro, también puede corregirse el ajustador o cambiarlos por otro que lo haya atendido.

The screenshot shows the 'ASISTE RADIO' application window. The title bar reads 'ASISTE RADIO'. The menu bar includes 'Controlador de Radio', 'Corrección de Tiempos', 'Asignación Ajustadores Dependencia', 'Trámite de Faltas', and 'Consulta de Cambios'. The 'Corrección de Tiempos' menu item is highlighted. Below the menu bar, there are input fields for 'Siniestro:' with the value '19689' and 'Ajustador:' with the value '476'. The main area is titled 'Descripción del Siniestro' and contains several input fields: 'Siniestro:', 'Fecha Siniestro:' (12/01/1998), 'Fecha de Reporte:' (12/01/1998), 'Póliza:', and 'Asegurador:'. To the right of these fields is a large empty box labeled 'Lugar del Siniestro:'. Below this section, there are two main sections: 'Ajustador:' and 'Tiempos:'. The 'Ajustador:' section has a dropdown menu with 'Ajustador' selected. The 'Tiempos:' section has four columns: 'Aviso', 'Arriba', 'Retiro del Asegurado', and 'Terminó'. Each column has a dropdown menu and a text input field. At the bottom of the window, a status bar shows 'Corrección de Tiempos a Siniestro', 'NUM', '18:58', and '21/01/1998'.

Asignación de Ajustadores de Dependencia

Algunos siniestros como los de Cristal o Robo o en ocasiones cuando el asegurado quiere ir a la delegación a terminar de atender el siniestro, en este caso se

cancela y se le asigna ajustador de Dependencias, por lo tanto cada delegación contiene ajustadores asignados los cuales atenderán a cualquier siniestro que se les presente. Al asignar un ajustador, automáticamente se le manda un mensaje por Skaytel para que tengan conocimiento de que persona va a llegar y los datos generales, tanto del auto, como del asegurado. En este caso también se pueden observar los siniestros que ya se han atendido y cuentan con un ajustador.

Siniestro:		Lugar del Siniestro:	
2755		UNIVERSIDAD ENO	
Fecha Siniestro:	10/04/1998	CONGRESO DEL FEDERAL	
Póliza:	1271041		
Incluc:	0000		
Asegurado:	01 SE BERTO MONTEPELLETIA		
Dependencia:		Ajustador:	
45 DEL ALVARO OBREGON			

Asignación: CAPS NUM ING 22:02 10/04/1998

Consulta de Cambios de Aseguradora

Esta última parte fue diseñada para poder consultar algunos siniestros que cambian de número de póliza, ya que al inicio cuando reportan el siniestros, no pueden encontrar los documentos necesarios, en este caso se crea una póliza fantasma, la cual se les asegura que se les atenderá debidamente, pero hay ocasiones que el asegurado pueda tener los papeles correctos y pueda dar el número de póliza correcto, en este caso se cambia al número de póliza correcta.

No Siniestro (Póliza Anterior)	Inicio Ast. (Póliza Actual)	Inicio Ast.	Situación del Siniestro	VIP	Nombre del A.
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					

2.2 COMPONENTES

Un componente (Servidor ActiveX) se considera como tal porque atiende a dos factores principales; su reusabilidad dentro del sistema o sistemas, y la dificultad de resolver en la estación de trabajo o la lógica que se requiere para cierto servicio.

Esto quiere decir, que si se desea resolver un requerimiento muy específico, que no se repite dentro del mismo sistema ni en ningún otro sistema, pero cuya lógica requiere de un desempeño mayor al de la estación de trabajo, el método utilizado para resolver este requerimiento se considera como un componente aunque no tenga reusabilidad.

La utilización de un mismo componente por varios servicios puede darse por **compartición** o **instanciación**. La compartición implica conectarse a un componente ya creado para utilizar sus servicios y la instanciación se refiere a crear una nueva instancia del componente cuando se requiera. Al diseñar un componente debe tenderse especialmente a la propiedad *instancing* de los Class Modules ya que esta propiedad define este comportamiento.

Los componentes serán creados mediante la función *CreateObject()* ya que al instanciarlos con *Dim comX as New COM* o con *Set comX = New COM* es necesario tener una referencia al componente COM dentro de la aplicación que crea la instancia, al hacer modificaciones al componente COM es necesario recompilar todos los módulos

que contengan esta referencia. En cambio CreateObject utiliza el ProgId (Componente.Clase) y no tiene referencias que actualizar.

Ejemplo: Se requiere crear conexiones RDO para cada consulta y actualización que se efectúe a la base de datos. Esta funcionalidad se utilizará en todos los servicios que tenga acceso a datos, por los que es conveniente pensar en ella como un componente, ya que al hacerle modificaciones, estas solo se harán en el código del componente y no en todas las funciones que lo implementen.

Los distintos servicios darán acceso al componente por instanciación ya que el desempeño debe ser rápido. El componente será un DLL que por definición (in – process) no puede compartirse.

La propiedad *instancing* de la clase que se encargue de crear la conexión dentro del componente será del tipo *MultiUse* ya que debe permitir el crear varias instancias de esta clase sin crear nuevas instancias de todo el componente.

2.2.1. TIPOS DE COMPONENTES

<i>Componente</i>	<i>Tipo</i>	<i>Pros</i>	<i>Contras</i>
DLL	In Process	Ejecución Rápida	Solo local No hay protección contra Caídas.

EXE	Local Out Process	Protección contra caídas	Mas lento que un DLL.
EXE	Remote Out Process	Ejecución Remota Se puede compartir Protección contra caídas	La velocidad de ejecución se ve afectada por el tráfico de Red.

La protección contra caídas se refiere a que un error irrecuperable que ocurra en un DLL ejecutándose en el mismo proceso de la aplicación que lo utiliza afectará también a la aplicación. Mientras que un EXE ejecutándose en su propio proceso no afecta a la aplicación.

2.2.2. SERVICIO DEL COMPONENTE.

El servicio que presta el componente debe ubicarse en la siguiente clasificación:

Componentes de Usuario: En su mayoría presentan interfaces gráficas o realizan validaciones que no implican un acceso a la base de datos. En caso de presentar interfaces gráficas estos componentes deben ubicarse en la estación de trabajo. En el caso de validaciones el componente solo estará en la estación de trabajo si esto se justifica con la necesidad de un mejor desempeño, pero por lo general deben encontrarse en el servidor.

Componentes de Negocio: Estos componentes son los que contienen las reglas de negocio, y se encargan de resolver la lógica de los requerimientos del usuario. Estos componentes son los que tienen más tendencia al cambio, por lo que generalmente deben ubicarse en el servidor con el fin de facilitar el mantenimiento al sistema. En estos componentes se preparan y ejecutan las consultas a la v base de datos y se resuelven fórmulas o algoritmos que requieran de una capacidad de procesamiento a la de la estación de trabajo.

Componentes de Datos: La gran mayoría de estos componentes está constituida por triggers y stored procedures ya que se encargan de la integridad referencial de los datos. Pero puede requerirse de un ActiveX que interactue directamente con la BDD mas allá de consultas o actualizaciones.

Estos tipos se dividen a su vez en **Generales** y **Particulares** atendiendo a lo siguiente.

Un componente que solo se utiliza dentro de un sistema será del tipo *Particular*, sin importar que se utiliza en uno o más módulos dentro del sistema en cuestión.

Un componente que se utilice en más de un sistema será del tipo *General*.

Tantos los componentes Generales como los Particulares pueden ser reutilizables, la clasificación se debe al ámbito de su utilización.

2.2.3. Ambito del Componente.

En esta sección se especifica todos los sistemas y módulos en los que se utiliza el componente con el fin de crear un *Mapa de Componentes* que se complementa mediante las clasificaciones previas para un *Diccionario de Componentes*.

Ejemplo: Para *istRDOManager* se mencionarán todos los módulos que hagan acceso a datos y el sistema al que pertenecen.

2.2.4. Descripción del Componente.

Se definirá el componente en base a los servicios que proporciona y explicará brevemente que tiene en común las clases que lo constituyen, y si no hay nada en común, él porque se agruparon en el mismo componente.

Ejemplo: *istRDOManager* proporciona herramientas para el manejo del ambiente RDO, y las clases que lo constituyen se encargan de proporcionar diferentes servicios en el manejo de este ambiente.

2.2.5. Clases del Componente.

Aquí se proporciona una lista de las clases que conforman el componente, mencionando el nombre, la instanciación (Propiedad *instancing*) y que otros componentes comparten esta clase, es decir que tienen agregado el archivo *.CLS*

correspondiente. Esto último se especifica con el fin de facilitar el mantenimiento de los componentes y añadir referencias a nivel clase dentro del mapa.

Ejemplo:

<i>Clase</i>	<i>Instancing</i>	<i>Compartido con ...</i>
RDOConnector	MultiUse	SmaSessionHandler
RDOEnvironmentManager	Private	

Las secciones siguientes aplican a cada una de las clases mencionadas en la lista anterior, las clases que se compartan con otros componentes tendrán su definición en el documento correspondiente a dicho componente y en la lista se especificará con un * el componente en cuestión.

2.2.5.1. Nombre de la Clase.

El nombre correspondiente a la propiedad *name* de la clase y el archivo .CLS en donde esta el código.

Ejemplo: RDOConnector, RDOConn.CLS

2.2.5.2. Descripción de la Clase.

Explicar el objetivo de la clase y porque agrupa los servicios que contiene.

Ejemplo: RDOConnector agrupa las funciones para establecer conexiones RDO a la BD, las distintas funciones asignan una conexión ya establecida o crean una nueva conexión para el usuario.

2.2.5.3. *Propiedades.*

En esta lista se indica si la propiedad está o no expuesta, el nombre breve descripción y el tipo de dato al que pertenece.

Nombre	Clasificación	Tipo	Descripción
RdoEnv	Pública	RDOEnvironment	Sesión del usuario en que se abrió la conexión

2.2.5.4. *Eventos*

Nombre y descripción del evento, parámetro de entrada y nombre del método o función en que se realiza el RaiseEvent.

2.2.5.5. *Componentes Agregados.*

Por cada componente que se agregue debe especificarse el nombre del componente al de la clase a utilizar, las propiedades y los métodos que se utilicen de esta clase. Cualquier componente instanciado dentro de nuestro componente es un componente agregado

2.2.6. Formas.

Nombre, archivo y descripción de las formas incluidas en el componente.

2.2.7. Datos Generales.

En esta última sección se anota el nombre de quien diseñó el componente, quien lo construyó; la versión actual del componente y la fecha en que se construyó la última versión

2.2.8. Componentes que se integraron al Módulo de Radio

Los componentes que se utilizaron para poder desarrollar el Módulo de Radio son dos, los cuales se explicarán su funcionamiento a continuación:

SnsSendMessage. Genera un archivo de texto el cual contiene información con respecto al siniestro, como el lugar donde ocurrió el siniestro, el tipo de vehículo, el tipo de siniestro, el nombre del conductor, si la póliza es vigente, es decir, si esta al corriente en los pagos, el nombre y clave del ajustador y el número de Pin del Skaytel y al final contiene el autor que genera el mensaje. Este archivo se almacena bajo cierta máquina en un subdirectorio fijo para que al momento que se genere un nuevo mensaje se envíe por medio de otro programa que corre bajo MS-DOS, el cual se trabaja por vía módem y hace llegar los mensajes a los ajustadores correspondientes.

Conecta. Como su nombre nos indica hace una conexión previa a la base de datos, por medio de un password y una clave de usuario, el cual permite tener un mejor control de la seguridad del sistema, una de las propiedades que contiene este componente es de poder llamar a otro programa ejecutable, sin que se pierdan las variables que se necesitan para poder llenar o tener los respectivos elementos al momento de llamar el programa correspondiente.

Estos componentes se tendrán que ejecutar antes de la ejecución del programa ya que están creados como objetos y contienen extensión .EXE, esto es para que se puedan llamar al momento de ejecución.

CAPITULO III

IMPLEMENTACION

Después de haber conocidos los prototipos y los componentes que se utilizan en el Módulo de Radio, nosotros podemos continuar con la programación y la instalación del sistema en las diferentes regiones, pero antes tenemos que conocer la base de datos que necesitamos para desarrollar los *queries*⁴ necesarios durante la programación del módulo.

3.1 BASE DE DATOS PARA EL DESARROLLO DEL MODULO DE RADIO.

Para desarrollar este modulo necesitaremos la consulta de varias tablas, esto es solo para la información más importante que necesita el radio-operador para darles la información correcta al ajustador y proceder con el servicio.

En este caso solo tomaremos en cuenta las tablas en donde se realizan las operaciones más importante de este módulo.

En la siguiente figura se muestra la parte correspondiente al módulo.

⁴ *Query*: Serie de ordencs para poder consultar, agregar y/o borrar: de una cierta base de datos de una o varias tablas

nosotros nos interesa la parte de ajustadores, ya que es donde encontramos la clave de usuario, la cual esta ligada con otra tabla importante “SN_ASIGNA_TURNO”, la cual podemos asignarles turnos, horarios y un estatus de habilitados para poder tomarlos al momento de asignar un ajustador a un siniestro.

“SN_ASIGNA_TURNO” también esta ligada con “SN_ZONA_BASE”, y esta se encuentra ligada con “SN_CV_REGIONAL”. Esto es porque cada empleado pertenece a un centro regional y cada región contiene varias zonas bases, las cuales son las bases de donde pertenecen los ajustadores y poder tener un mayor control.

“SN_CITA” es otra tabla importante, aquí tenemos información sobre las citas que se realizan para algún siniestro de robo o cristal, en esta parte se manejará dentro del botón de CITA, ya que en esa parte podremos observar todas las citas pendientes que tiene dicho siniestro y también se pueden cancelar en esa parte.

Las tablas faltantes solo se toman en cuenta en las consultas para tener de referencia la ubicación del siniestro (SN_SINIESTRO), el número de póliza y tipo de póliza (SN_SINI_POLIZA, SN_POLIZA).

3.2 PROGRAMACIÓN DE PROTOTIPOS

Comenzaremos con la seguridad del sistema, en esta parte se utilizó una librería llamada KeyAdmin, la cual se instancia en el "load" de la forma con el siguiente código:

```
Set udtConecta = CreateObject("Conexión.IUtilityClass")
Set keyAdmin = CreateObject("KeyClientRDO.KeyClientRDO.1")

' ***** Control del KeyAdmin *****
If keyAdmin.DlgLogin("ASISTE", "", 0) = 1 Then
    If keyAdmin.Validate(106000) = k_ok Then
        If keyAdmin.Validate(100000) = k_ok Then
            If udtConecta.Conectar(Cnet, keyAdmin.sDBName, keyAdmin.sDBUser,
                keyAdmin.sDBPwd) Then
                MsgBox "Conexión lograda"
                FrmRadio.Caption = GSTR_SLS_SISTEMA + " - R A D I O"
            Else
                MsgBox "Por el momento no hay conexión a la BD"
            End If
        End If
    Else
        MsgBox "No tienes acceso a este sistema"
    End If
Else
    MsgBox " no tienes acceso a este módulo"
End If
Else
    MsgBox "No tienes acceso"
End If
End If
```

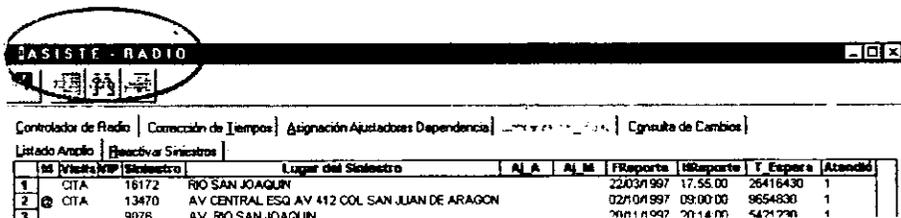
En donde las dos primeras líneas son la instanciación de componente de conexión y *udtConecta* es un objeto de conexión y *keyadmin* es un objeto del componente KeyAdmin que se encuentra dentro del componente conexión.

El primer "IF" que se encuentra es para asegurarse que el "login" del usuario se encuentre dentro del proyecto llamado "ASISTE".

El segundo "IF" valida el "login" sea correcto al igual que el tercer "IF", solo que con el password. El Cuarto "IF" checar que la variable udtConecta haga una conexión a la base de datos, en donde Cnet, es la conexión, "keyadmin.sDBname" es el nombre de la base de datos, "KeyAdmin.sDBUser" es el username o conocido como login y "keyAdmin.sDBPwd" es el password.

Al tener todo como verdadero podemos tener una conexión asegurada y podemos mandar a llamar nuestra forma.

Al tener nuestra conexión, tenemos una instrucción `FrmRadio.Caption = GSTR_SLS_SISTEMA + " - R A D I O"` en donde `FrmRadio.Caption` es la forma de nuestro sistema, "GSTR_SLS_SISTEMA" es una constante que se encuentra dentro de un modulo (.bas) y contiene el nombre del sistema ("ASISTE"), el cual aparece en nuestra forma al inicio de nuestro programa como se ilustra en la siguiente figura dentro del círculo rojo:



Continuaremos con el Query que se utiliza para mostrar la información del listado amplio.

```
SQL = "SELECT DISTINCT p.cve_vip, s.no_siniestro, s.lugar, "
SQL = SQL + " s.fec_reporte, s.cve_usr_atendio "
SQL = SQL + " From sn_poliza p, sn_siniestro s, sn_zona_base sb, "
SQL = SQL + " sn_sini_poliza sp, sn_sinies_vehí sv "
SQL = SQL + " WHERE s.no_siniestro = sp.no_siniestro "
SQL = SQL + " AND p.cve_aseguradora = sp.cve_aseguradora "
SQL = SQL + " AND p.poliza = sp.poliza "
SQL = SQL + " AND p.inciso = sp.inciso "
SQL = SQL + " AND s.no_siniestro = sv.no_siniestro "
SQL = SQL + " AND s.cve_cv = sb.cve_cv "
SQL = SQL + " AND sv.cve_base_sini = sb.cve_base "
SQL = SQL + " AND ((sb.es_dependencia='C') "
SQL = SQL + " AND (s.cve_estatus=1) "
SQL = SQL + " AND (s.cve_cv=' & CveCV & ") "
SQL = SQL + " AND ((s.cve_lugar_aten=23003) OR "
SQL = SQL + " (s.cve_lugar_aten=23004))) "
SQL = SQL + " ORDER BY s.fec_reporte"
```

El Query anterior es un query múltiple el cual relaciona cinco tablas, de las cuales se extrae la información de los siniestros que se encuentran en atención. (Como se muestra en la siguiente figura dentro del cuadro rojo)

Vista/Vip	Siniestro	Lugar del siniestro	A/A	A/A	Transporte	Transporte	Espera (Atendido)
1	CITA 16172	RIO SAN JOAQUIN	22/03/1987	17:55:00	26416430		1
2	CITA 13470	AV CENTRAL ESQ AV 412 COL SAN JUAN DE ARAOON	02/04/1987	09:00:00	9654830		1
3	4076	AV. RIO SAN JOAQUIN	20/11/1987	20:14:00	5421720		1

Al momento que aparece un siniestro en el listado amplio se debe de asignar inmediatamente un ajustador, solo en caso de que no haya alguno libre, este se deja pendiente hasta que haya alguno libre o se habilite. El siguiente proceso es la programación que realiza todo esto proceso:

```

Private Sub FBkListAmp_Click(ByVal nRow As Long, ByVal nCol As Long)
FBkListAmp.MousePointer = 11
StsBar.Panels(1).Text = "Espere un momento por favor ..."
gRow = FBkListAmp.Row
If FBkListAmp.TextRC(gRow, 4) <> "" Then `si # de siniestro <> ""
LimpiaMascara ` limpia las mascaras del listado amplio

gRow = FBkListAmp.Row `Toma el número del renglón elegido
txtAjusAuto = FBkListAmp.TextRC(gRow, 6)
txtAjusMoto = FBkListAmp.TextRC(gRow, 7)
FrmLibera.Cual_Ajus
If txtAjusAuto = "" And txtAjusMoto = "" Or txtAjusAuto = " _
" Or txtAjusMoto = " " Then `Si no tiene ajustador _
CmdCancelaSini.Enabled = True `Habilita botón de cancelación
CmdLiberaAj.Enabled = False `Deshabilita botón de liberación
Else
CmdCancelaSini.Enabled = False `Deshabilita botón de
`cancelación.
CmdLiberaAj.Enabled = True `Habilita botón de liberación
End If
If txtAjusAuto <> "" Then
If txtAjusAuto <> " " Then
CmdMensAjA.Enabled = True
Else
CmdMensAjA.Enabled = False
End If
Else
CmdMensAjA.Enabled = False
End If
If txtAjusMoto <> "" Then
If txtAjusMoto <> " " Then
CmdMenAjM.Enabled = True
Else
CmdMenAjM.Enabled = False
End If
Else
CmdMenAjM.Enabled = False
End If
NoSiniestro = FBkListAmp.TextRC(gRow, 4) `Toma el número de
` siniestro
Cita0 = FBkListAmp.TextRC(gRow, 2)
If Cita0 = "CITA" Then `Si el número
` de siniestro tiene cita, habilita el botón de visita
CmdVisitas.Enabled = True
Else
CmdVisitas.Enabled = False
End If
On Error GoTo 0
Ajustadores NoSiniestro, 24002 `Llama al procedimiento
` Ajustadores para asignar ajustador
Ajustadores NoSiniestro, 24001 `o tomar el ajustador que ya
` se selecciono
StsBar.Panels(1).Text = "No. de Siniestro en Atención: " + _
NoSiniestro

```

```

Else
  StsBar.Panels(1).Text = "No. de Siniestro en Atención: " + _
  NoSiniestro
End If
Else
  LimpiaMascara
  MsgBox "NO SE HA SELECCIONADO SINIESTRO"
  StsBar.Panels(1).Text = "Asigna Tiempos de Atención de _
  Siniestros en Crucero"
End If
FBkListAmp.MousePointer = 1
Toolbar1.Buttons(7).Enabled = True
End Sub

```

En el código anterior se activa con el evento CLICK del grid el proceso que contiene este evento consiste en verificar si tiene algún ajustador, si encuentra alguno habilita el botón de liberar ajustador y el botón de mensaje, dependiendo del ajustador que este asignado y deshabilita el botón de cancelación. En el caso de que no tenga ajustadores habilita el botón de cancelación y deshabilita los botones de mensaje y liberación de ajustador.

Posteriormente se ejecuta el procedimiento *Ajustadores (No_Siniestro, Tipo_Ajustador)* donde el *No_Siniestro* es el número de siniestro en que se encuentra posicionado el cursor en el grid y *Tipo_Ajustador* es el tipo de ajustador en este caso contamos con dos tipos el 24001 pertenece a un ajustador de moto y el 24002 para el ajustador de auto. Si tiene ajustador asignado solo muestra el número y nombre del ajustador en la parte de abajo con sus respectivas horas, esto lo hace para los dos tipos de ajustadores, si no se encuentra uno de ellos o en el caso de que no haya ninguno se traen todos los ajustadores que se encuentren libres, dependiendo de la zona en donde se

encuentre el siniestro o en caso contrario se traen todos de todas las zonas que pertenezcan al centro de valuación⁵.

```
Public Function Ajustadores(gRow1 As String, tipo_ajus As String)
Dim noempleado As Long
Dim Aviso As String
Dim Llegada As String
Dim Termino As String
Dim Retiro As String

'Query para verificar que existe el número de siniestro

strSQL1 = "Select sn_asigna_ajust.no_empleado,
strSQL1 = "sn_asigna_ajust.fec_aviso,"
strSQL1 = strSQL1 + "sn_asigna_ajust.fec_llegada,
strSQL1 = "sn_asigna_ajust.fec_termino, "
strSQL1 = "sn_asigna_ajust.fec_retiro_aseg"
strSQL1 = strSQL1 + " From sn_asigna_ajust sn_asigna_ajust,
strSQL1 = "sn_siniestro sn_siniestro"
strSQL1 = strSQL1 + " Where sn_asigna_ajust.no_siniestro = '" & _
gRow1 & "'"
strSQL1 = strSQL1 + " AND sn_siniestro.no_siniestro = _
sn_asigna_ajust.no_siniestro"
strSQL1 = strSQL1 + " AND ((sn_asigna_ajust.cve_tipo_ajus= '" & _
tipo_ajus & "')"
strSQL1 = strSQL1 + " AND (sn_asigna_ajust.cve_est_aten <> 26003))"
Set rs = Cnet.OpenResultset(strSQL1, 0, rdConcurReadOnly)

If rs.EOF And rs.BOF Then
If tipo_ajus = 24002 Then
Asig_Ajustador tipo_ajus, CboAjusAuto
K = K + 1
End If
If tipo_ajus = 24001 Then
Asig_Ajustador tipo_ajus, CboAjusMoto
K = K + 1
End If
If tipo_ajus = 24003 Then
Asig_Ajustador tipo_ajus, CmbDependencia
CmbDependencia.Enabled = True
K = K + 1
End If
Else
If tipo_ajus = 24002 Then
noempleado = rs(0)
Aviso = ChecarNulls(rs(1))
Llegada = ChecarNulls(rs(2))
Termino = ChecarNulls(rs(3))
Retiro = ChecarNulls(rs(4))
Empleado noempleado, CboAjusAuto
```

⁵ Centro de Valuación (CV): Lugar de donde pertenecen los empleados, ajustadores, etc., por ejemplo CV=1 quiere decir que pertenece a México Local (Lago Xochimilco)

```

CboAjusAuto.Enabled = False
CmdAvisoA(0).Enabled = False
If Aviso <> "" Then
    MaskedTextBox2(8) = Format$(CDate(Aviso), "hh:mm:ss")
End If
CmdArriboA(0).Enabled = True
If Llegada <> "" Then
    CmdArriboA(0).Enabled = False
    MaskedTextBox2(4) = Format$(CDate(Llegada), "hh:mm:ss")
    CmdRetiroA(0).Enabled = True
    CmdTerminoA(0).Enabled = True
End If
If Retiro <> "" Then
    CmdRetiroA(0).Enabled = False
    MaskedTextBox2(5) = Format$(CDate(Retiro), "hh:mm:ss")
    CmdTerminoA(0).Enabled = True
    CmdRetiroM(1).Enabled = False
    MaskedTextBox2(10).Text = MaskedTextBox2(5)
End If
If Termino <> "" Then
    CmdTerminoA(0).Enabled = False
    MaskedTextBox2(6) = Format$(CDate(Termino), "hh:mm:ss")
    FrmAjAuto.Enabled = False
End If
End If

If tipo_ajus = 24001 Then
    noempleado = rs(0)
    Aviso = ChecarNulls(rs(1))
    Llegada = ChecarNulls(rs(2))
    Termino = ChecarNulls(rs(3))
    Retiro = ChecarNulls(rs(4))
    Empleado noempleado, CboAjusMoto
    CmdAvisoM(1).Enabled = False
    CboAjusMoto.Enabled = False
    If Aviso <> "" Then
        MaskedTextBox2(7) = Format$(CDate(Aviso), "hh:mm:ss")
        CmdArriboM(1).Enabled = True
    End If
    If llegada <> "" Then
        CmdArriboM(1).Enabled = False
        MaskedTextBox2(9) = Format$(CDate(Llegada), "hh:mm:ss")
        CmdRetiroM(1).Enabled = True
        CmdTerminoM(1).Enabled = True
    End If
    If Retiro <> "" Then
        CmdRetiroM(1).Enabled = False
        MaskedTextBox2(10) = Format$(CDate(Retiro), "hh:mm:ss")
        CmdTerminoM(1).Enabled = True
        MaskedTextBox2(5) = MaskedTextBox2(10)
        CmdRetiroA(0).Enabled = False
    End If
    If MaskedTextBox2(5).Text <> "__:__:__" Then
        CmdRetiroM(1).Enabled = False
    End If
    If Termino <> "" Then
        CmdTerminoM(1).Enabled = False
    End If

```

```

        MaskEdBox2(11) = Format$(CDate(Termino), "hh:mm:ss")
        FrmAjMoto.Enabled = False
    End If
End If

If tipo_ajus = 24003 Then
    noempleado = rs(0)
    Empleado noempleado, CmbDependencia
    CmbDependencia.Enabled = False
End If

End If

End Function

```

En este Código observamos que en primer lugar realizamos un *query* en donde traemos toda la información del ajustador asignado al siniestro que sé esta consultando. Al momento de ejecutarlo y si trae los datos correspondientes proseguimos con la colocación de los datos. En caso contrario se ejecuta otro procedimiento llamado *Asig_Ajustador (Tipo_Ajustador, ComboBox)* donde *Tipo_Ajustador* es el tipo de ajustador que se necesita (Moto o Auto) y *ComboBox* es el objeto en donde se van insertar los ajustadores que se encuentren disponibles.

El procedimiento *Ajustadores*, también nos sirve para asignar ajustadores de Dependencia, esto es para el caso de Asignación de Ajustadores de Dependencia, ya que verifica si ya contiene asignado algún ajustador y si es así solo coloca el nombre del ajustador en el ComboBox correspondiente, en caso contrario regresarán todos los ajustadores de dependencia para la asignación de uno de ellos.

Ahora procederemos a verificar el procedimiento *Asig_Ajustador*:

```
Public Sub Asig_Ajustador(tipo_ajus As String, combo As ComboBox)
    Dim ajustadl() As String
    Dim ajt As String
    Dim kk As Boolean
    Dim Nombre As String
    Dim Paterno As String
    Dim Materno As String
    Dim blnBan As Boolean

    blnBan = False
    ' **** Tipo de Siniestro ****
    strSQL = "Select A.cve_tsiniestro from sn_siniestro A Where _
no_siniestro=" & NoSiniestro & ""
    Set rsi = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
    ' **** Ajustadores ****
    strSQL = " SELECT DM.cve_ajustador, DM.nombre, DM.a_paterno, _
DM.a_materno "
    strSQL = strSQL + " FROM sn_asigna_turno SAT, sn_directorio_mul _
DM, sn_zona_base ZB "
    strSQL = strSQL + " WHERE SAT.no_empleado = DM.cve_usuario "
    strSQL = strSQL + " AND SAT.cve_base = ZB.cve_base "
    strSQL = strSQL + " AND ((ZB.cve_cv=1) "
    strSQL = strSQL + " AND (SAT.dia_semana=" & Format$(PCurrent, "w")
+ ") "
    strSQL = strSQL + " AND (SAT.cve_tipo_ajus=" & tipo_ajus & ") "
    strSQL = strSQL + " AND (DM.habilitado = 'S'))"
    If ChecarNullsNum(rsl{0}) = 10 Then
        strSQL = strSQL + " AND ZB.cve_base=80 "
    End If
    strSQL = strSQL + " AND SAT.fec_inicio <= current "
    strSQL = strSQL + " AND current <= SAT.fec_fin "
    strSQL = strSQL + " AND '" & Format$(PCurrent, "hh:mm") & "' _
between SAT.hora_inicial AND SAT.hora_final "
    Set rs = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
    ReDim Ajustad(100) As String
    JI = 1
    If rs.EOF And rs.BOF Then
        Ajus_Otros tipo_ajus, combo
        MousePointer = 0
        Exit Sub
        On Error GoTo 0
    Else
        combo.Enabled = True
        If tipo_ajus = 24003 Then
            While Not rs.EOF
                combo.AddItem ChecarNulls(rs(0)) + " - " + _
ChecarNulls(rs(1)) + " " + ChecarNulls(rs(2)) + " " + _
ChecarNulls(rs(3))
                rs.MoveNext
            Wend
            Exit Sub
        Else
            While Not rs.EOF
```

```

        Ajustad(JI) = rs(0) + " - " + ChecarNulls(rs(1)) + " " +
        ChecarNulls(rs(2)) + " " + ChecarNulls(rs(3),
        rs.MoveNext
        JI = JI + 1
    Wend
End If
End If

strSQL = "SELECT DISTINCT sn_directorio_mul.cve_ajustador "
strSQL = strSQL + "From sn_asigna_ajust sn_asigna_ajust, _
        sn_directorio_mul sn_directorio_mul "
strSQL = strSQL + "Where sn_asigna_ajust.No_Empleado = _
        sn_directorio_mul.cve_usuario "
strSQL = strSQL + "And ((sn_asigna_ajust.cve_tipo_ajus = " & _
        tipo_ajus & " ) "
strSQL = strSQL + "And (sn_asigna_ajust.cve_est_aten = 26001) "
strSQL = strSQL + "And (sn_directorio_mul.cve_cv= " & CveCV & ")))"
Set rs = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
ReDim ajustadl(100) As String
JK = 1
If Not (rs.EOF And rs.BOF) Then
    While Not rs.EOF
        ajustadl(JK) = rs(0)
        JK = JK + 1
        rs.MoveNext
    Wend
End If
For IK = 1 To JI - 1
    For IJ = 1 To JK - 1
        cosa = Mid(Ajustad(IK), 1, 5)
        If cosa = ajustadl(IJ) Then
            kk = True
            Exit For
        Else
            kk = False
        End If
    Next
    If kk = False Then
        combo.AddItem Ajustad(IK)           'Llena el combo seleccionado
        blnBan = True
    End If
    If blnBan = False Then
        Ajus_Otros tipo_ajus, combo
        MousePointer = 0
        Exit Sub
    End If
Next
MousePointer = 0
End Sub

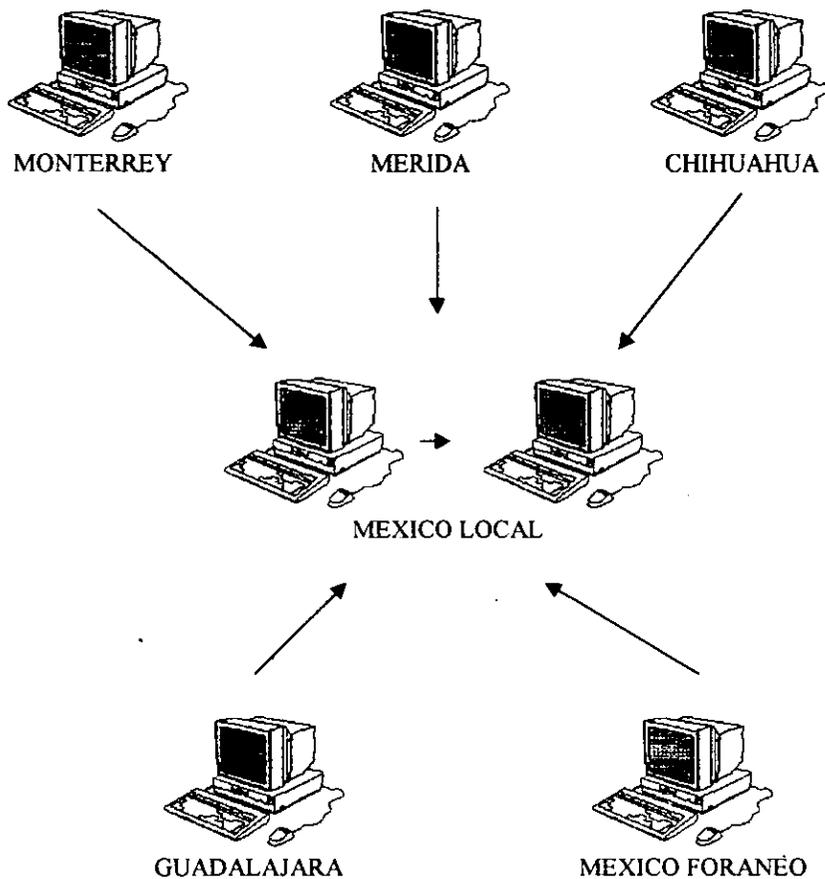
```

Como podremos ver primeramente realizamos un *query* en donde traemos el tipo de siniestro que estamos atendiendo, ya que al conocerlo se enviará un ajustador para este tipo de siniestro, en caso de que no haya ajustadores por tipo de siniestro posteriormente realizaremos otro *query* en donde regresen todos los ajustadores de todos los tipos de siniestro y se envían al ComboBox para su elección.

El desarrollo del programa completo de encontrarán en el anexo 1.

3.3 INSTALACION.

Para la instalación del proyecto ASISTE se necesitaron tener un servidor por cada división, como se muestra en la siguiente figura.



En cada uno de estos servidores tendrán una copia del proyecto y de los componentes que se utilizan, estos contarán con un sistema de Windows NT Server, esto es para la facilidad de la ejecución de los programas y los manejos de red.

En cada una de las divisiones tendremos como mínimo una terminal por módulo, en el caso del módulo de Radio tendremos como mínimo una terminal y como máximo tendremos tres terminales, esto es en el caso de México Local, ya que es donde surgen más siniestros que en las demás divisiones.

Estas terminales contarán con un sistema de Windows 95 y contarán como mínimo 16 MB en Ram a 133 Mhertz, esto es para que no dificulte la ejecución de los programas, además no entorpezca con la velocidad de la red.

Para en el caso de la base de datos contaremos con un servidor que contendrá un sistema Unix, el cual podrá almacenar mas de 10 GB de información, el cual estará programado que a determinada cantidad de información, esta pasará la información a otro servidor de respaldo y como historial contendremos la información más antigua.

En caso de que se caiga el sistema entrará otro servidor con las mismas especificaciones, el cual se está actualizando día con día para contener la misma información, ya que en caso de caídas no se pierda la nueva información que se almacene en ese instante.

CONCLUSIONES

El proyecto “Sistema Integral de Indemnización a Siniestros Autos de Seguros Monterrey Aetna”, es el proyecto más importante que contiene Seguros Monterrey Aetna, el cual busca rapidez e integración de documentos para darle un mejor servicio a sus clientes. Por tal motivo algunas condiciones de infraestructura ya estaban dadas al inicio del proyecto, como era el software que se utilizaría y el tiempo en que se debería de terminar. En este caso el Modulo de Radio es el proceso más importante para el proyecto, ya que sin él no se podría dar seguimiento a los siniestros.

Multiasistencia es un área de Seguros Monterrey Aetna que se encarga de administrar y atender todos los siniestros que se puedan presentar en las 24 horas del día y atender lo mejor y más rápido posible al cliente que adquiere uno de sus seguros, por tal motivo los requerimientos e inconformidades que constantemente manifestaban problemas de integridad, dificultad de seguimiento, carencia de reportes específicos y generales y algunas más, se torno difícil su labor y con pobres rendimientos, accesorios y servicios.

Por eso antes de llevar a cabo el desarrollo de un sistema de información se tiene que establecer claramente los objetivos a alcanzar y las personas encargadas del desarrollo de la aplicación, deben comprenderlos y tenerlos siempre presente en la realización del proyecto. Esto con el fin de poder determinar de una manera más práctica y sin que se modifiquen los objetivos, las necesidades reales de los usuarios, ya que

algunas ocasiones los usuarios no saben exactamente lo que necesitan y otros más ni siquiera expresan una necesidad.

También aprendí que el analista debe conocer lo mejor posible el proceso a analizar antes de entrevistarse con el usuario para recabar mayor información, pues debe saber que preguntas hacer, que recomendaciones dar y hacia donde debe guiar la investigación, ya que el éxito del proyecto depende de la buena comunicación con el usuario, pues nadie mejor que ellos conoce el ámbito donde trabajará el sistema.

Toda esta labor nos permitió dar la mejor solución al problema dadas las restricciones de fechas de entrega, los límites presupuestales, la disponibilidad del personal y las interfaces técnicas, además depende de un buen análisis y un diseño para poder tener un sistema eficiente y confiable, en pocas palabras, con calidad.

Sin embargo, y a pesar de todos los obstáculos a los que nos enfrentamos se logró la terminación de un sistema que cumple con todos los objetivos propuestos y uno de ellos es la calidad y eficiencia de atención a los clientes.

Una ventaja que presenta el sistema SIISA es el número de usuarios que lo pueden operar, ya que presenta concurrencia en cada parte importante para que el sistema sea seguro y no se encuentren datos repetidos en los momentos de las consultas.

El tiempo de atención y ayuda a los ajustadores es más rápida, esto hace que el asegurado o el cliente se le atiende más rápidamente en los momentos de siniestro.

Una desventaja que se presenta en el sistema es que su correcta operación depende en gran medida del buen funcionamiento de la red, pues si la comunicación entre los equipos en la red es lenta, la búsqueda que se realiza el sistema toma más tiempo. Esto hace necesario llevar a cabo un continuo monitoreo de la red para verificar que la velocidad de transmisión sea la correcta o en caso contrario se tendría que tener un ancho de banda más grande para que la información fluya sin ningún problema.

Por otra parte, esta investigación y desarrollo sirva de base para la expansión de alcance del sistema, en el sentido de brindarles más apoyo al área de Multiasistencia y dar un mejor servicios a todos los clientes de Seguros Monterrey.

ANEXO I

Public Sub Tiempo_Arribo()

```

'* PROPOSITO      : Checa si no se le ha asignado Tiempo de arribo es correcto  *
'*                                                       *
'* PARAMETROS    :                                                       *
'*                                                       *
'* REGRESO       :                                                       *
'*                                                       *

Dim intI As Integer
Dim strSin As String
Dim intRow As Integer
Dim dtmFecLlegaA As Date
Dim dtmFecLlegaM As Date
Dim dtmFecLlega As Date
Dim strFecLlega As String
Dim strDate_Timel As String
Dim rstTime As String
Dim strAuto As String
Dim strMoto As String

intRow = FBkListAmp.MaxRow
intI = 1

Do While intI <= intRow
    strSin = Trim(FBkListAmp.TextRC(intI, 4))
    strAuto = Trim(FBkListAmp.TextRC(intI, 6))
    strMoto = Trim(FBkListAmp.TextRC(intI, 7))
    If strAuto <> "" Then
        Auto = True
    Else
        Auto = False
    End If
    If strMoto <> "" Then
        Moto = True
    Else
        Moto = False
    End If
    If Auto = True Then
        strSQL = "SELECT A.fec_llegada "
        strSQL = strSQL + "From sn_asigna_ajust A "
        strSQL = strSQL + "WHERE (A.no_siniestro='" & strSin & "'"
        strSQL = strSQL + "AND (A.cve_est_aten=2600); AND (A.cve_tipo_ajus=24002)"
        Set rs = Cnet.OpenResultset(strSQL, rdOpenKeyset, rdConcurRowVer)
        If Not (rs.EOF And rs.BOF) Then
            If IsNull(rs(0)) Then
                dtmFecLlegaA = "00:00:00"
            Else
                dtmFecLlegaA = rs(0)
            End If
        End If
    End If
    If Moto = True Then
        strSQL = "SELECT A.fec_llegada "
        strSQL = strSQL + "From sn_asigna_ajust A "
        strSQL = strSQL + "WHERE (A.no_siniestro='" & strSin & "'"
        strSQL = strSQL + "AND (A.cve_est_aten=2600) AND (A.cve_tipo_ajus=24001)"
        Set rs = Cnet.OpenResultset(strSQL, rdOpenKeyset, rdConcurRowVer)
        If Not (rs.EOF And rs.BOF) Then
            If IsNull(rs(0)) Then
                dtmFecLlegaM = "00:00:00"
            Else
                dtmFecLlegaM = rs(0)
            End If
        End If
    End If
    If Auto = True And Moto = True Then
        strDate_Timel = FBkListAmp.EntryRC(intI, 8) & " " & FBkListAmp.EntryRC(intI,
9)

        If dtmFecLlegaA <> "00:00:00" And dtmFecLlegaM <> "00:00:00" Then
            If dtmFecLlegaA < dtmFecLlegaM Then
                If strDate_Timel <> "" Then

```

```

        rstTime = DateDiff("n", strDate_Timel, dtmFecLlegaA)
        FBkListAmp.TextRC(intI, 10) = rstTime
    End If
Else
    If strDate_Timel <> "" Then
        rstTime = DateDiff("n", strDate_Timel, dtmFecLlegaM)
        FBkListAmp.TextRC(intI, 10) = rstTime
    End If
End If
Else
    If dtmFecLlegaA <> "00:00:00" Then
        If strDate_Timel <> "" Then
            rstTime = DateDiff("n", strDate_Timel, dtmFecLlegaA)
            FBkListAmp.TextRC(intI, 10) = rstTime
        End If
    ElseIf dtmFecLlegaM <> "00:00:00" Then
        If strDate_Timel <> "" Then
            rstTime = DateDiff("n", strDate_Timel, dtmFecLlegaM)
            FBkListAmp.TextRC(intI, 10) = rstTime
        End If
    End If
End If
Else
    If Auto = True Then
        If dtmFecLlegaA <> "00:00:00" Then
            strDate_Timel = FBkListAmp.EntryRC(intI, 8) & " " &
FBkListAmp.EntryRC(intI, 9)
            If strDate_Timel <> "" Then
                rstTime = DateDiff("n", strDate_Timel, dtmFecLlegaA)
                FBkListAmp.TextRC(intI, 10) = rstTime
            End If
        End If
    End If
    If Moto = True Then
        If dtmFecLlegaM <> "00:00:00" Then
            strDate_Timel = FBkListAmp.EntryRC(intI, 8) & " " &
FBkListAmp.EntryRC(intI, 9)
            If strDate_Timel <> "" Then
                rstTime = DateDiff("n", strDate_Timel, dtmFecLlegaM)
                FBkListAmp.TextRC(intI, 10) = rstTime
            End If
        End If
    End If
End If
    intI = intI + 1
Loop
End Sub

```

Public Function Query_Checa() As Integer

```

' * PROPOSITO      : Regresa el número de siniestros que estan en atención      *
' * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
' * PARAMETROS    : * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
' * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
' * REGRESO       : Query_Checa as Integer *
' * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
SQL = "SELECT DISTINCT count(*) "
SQL = SQL + " From sn_poliza p, sn_siniestro s, sn_zona_base sb, sn_sini_poliza sp,
sn_sinies_vehí sv "
SQL = SQL + " WHERE s.no_siniestro = sp.no_siniestro "
SQL = SQL + " AND p.cve_aseguradora = sp.cve_aseguradora "
SQL = SQL + " AND p.poliza = sp.poliza "
SQL = SQL + " AND p.inciso = sp.inciso "
SQL = SQL + " AND s.no_siniestro = sv.no_siniestro "
SQL = SQL + " AND s.cve_cv = sb.cve_cv "
SQL = SQL + " AND sv.cve_base_sini = sb.cve_base "
SQL = SQL + " AND ((sb.es_dependencia='C') "
SQL = SQL + " AND (s.cve_estatus=1) "
SQL = SQL + " AND (s.cve_cv=" & CveCV & ") "
SQL = SQL + " AND ((s.cve_lugar_aten=23003) OR (s.cve_lugar_aten=23001)) "

```

```

'SQL = SQL + " ORDER BY s.fec_reporte"
Set rs = Cnet.OpenResultset(SQL, rdOpenKeyset, rdConcurRowVer)
If Not (rs.EOF And rs.BOF) Then
    Query_Checa = ChecarNullsNum(rs(0))
Else
    Query_Checa = 0
End If
End Function

Public Sub Query_Conecta()
'* PROPOSITO      : Contiene la información de los siniestros que se encuentran *
'*               : en estatus de atención *
'*               : *
'* PARAMETROS    : *
'*               : *
'* REGRESO       : *
'*               : *

On Error GoTo Err_Conecta

    SQL = "SELECT DISTINCT p.cve_vip, s.no_siniestro, s.lugar, s.fec_reporte,
s.cve_usr_atendio "
    SQL = SQL + " From sn_poliza p, sn_siniestro s, sn_zona_base sb, sn_sini_poliza sp,
sn_sinies_vehi sv "
    SQL = SQL + " WHERE s.no_siniestro = sp.no_siniestro "
    SQL = SQL + " AND p.cve_aseguradora = sp.cve_aseguradora "
    SQL = SQL + " AND p.poliza = sp.poliza "
    SQL = SQL + " AND p.inciso = sp.inciso "
    SQL = SQL + " AND s.no_siniestro = sv.no_siniestro "
    SQL = SQL + " AND s.cve_cv = sb.cve_cv "
    SQL = SQL + " AND sv.cve_base_sini = sb.cve_base "
    SQL = SQL + " AND ((sb.es_dependencia='C') "
    SQL = SQL + " AND (s.cve_estatus=1) "
    SQL = SQL + " AND (s.cve_cv=" & CveCV & ") "
    SQL = SQL + " AND ((s.cve_lugar_aten=23003) OR (s.cve_lugar_aten=23004))) "
    SQL = SQL + " ORDER BY s.fec_reporte"
    Set rs = Cnet.OpenResultset(SQL, rdOpenKeyset, rdConcurRowVer)
Exit Sub
Err_Conecta:
'MsgBox Err.Number & "-" & Err.Description
Dim er As rdoError
Dim lngerror As Long, intResp As Integer
For Each er In rdoErrors
    msg = msg & er.Description
    & "." & er.Number & vbCr
    If er.Number = -144 Then 'Lock en un Registro de la tabla.
        lngerror = er.Number
    Exit For
End If
Next er
If lngerror = -144 Then 'Lock en un Registro de la tabla.
'MsgBox "Se ejecutará nuevamente la Consulta. ", vbInformation
Resume
End If
End Sub

Public Sub Tiempo_LA()
'* PROPOSITO      : Obtiene el tiempo de espera y lo coloca en el grid del *
'*               : listado amplio *
'*               : *
'* PARAMETROS    : lngEmpleado, strSiniestro *
'*               : *
'* REGRESO       : Checa_Siniestro as Boolean *
'*               : *

Dim strDate_Timel As String
Dim rstTime As String
Dim intI As Integer
Dim intK As Integer

```

```

Dim intRow As Integer

intRow = FBkListAmp.MaxRow
intI = 1
Do While intI <= intRow
    strDate_Timel = Trim(FBkListAmp.EntryRC(intI, 8) & " " & FBkListAmp.EntryRC(intI,
9))
    If strDate_Timel <> "" Then
        rstTime = DateDiff("n", strDate_Timel, PCurrent)
        FBkListAmp.TextRC(intI, 11) = rstTime
        intI = intI + 1
        intK = intK + 1
    Else
        Exit Do
    End If
Loop
End Sub

```

Public Sub Conecta()

```

'* PROPOSITO      : Conecta y refresca la base de datos
'*
'* PARAMETROS    :
'*
'* REGRESO       :
'*

```

```

Dim intEN As Integer
Dim intintRowG As Integer
Dim Fals As Boolean
Dim Tru As Boolean
Dim Tiempo As String
Dim Sn As String
Dim intMR As Integer
Dim intRow As Integer
Dim intI As Integer
Dim Date_Timel As String
ColG = 3
intintRowG = 0
Fals = False
Tru = True

Query_Conecta
If Not (rs.EOF And rs.BOF) Then
    While Not rs.EOF
        introwg = introwg + 1
        For intEN = 0 To 2
            If intEN = 0 Then
                If ChecarNulls(rs(intEN)) <> "" Then
                    FBkListAmp.TextRC(introwg, ColG + intEN) = "&"
                End If
            Else
                If intEN = 2 Then
                    FBkListAmp.TextRC(introwg, ColG + intEN) =
Mid(ChecarNulls(rs(intEN)), 1, 44)
                Else
                    FBkListAmp.TextRC(introwg, ColG + intEN) = ChecarNulls(rs(intEN))
                End If
            End If
        Next
        If intEN = 3 Then
            Tiempo = Format$(rs(intEN), "dd/mm/yyyy hh:mm:ss")
            fecc = Mid(Tiempo, 1, 10)
            horr = Mid(Tiempo, 12, 8)
            FBkListAmp.TextRC(introwg, 8) = fecc
            FBkListAmp.TextRC(introwg, 9) = horr
            intEN = intEN + 1
            FBkListAmp.TextRC(introwg, 12) = rs(intEN)
        End If
        rs.MoveNext
    End While
End Sub

```

```

Wend

    strSQL = "SELECT sn_cita.no_siniestro"          ' Checa los siniestros que
tengan citas
    strSQL = strSQL + " From sn_cita sn_cita"
    strSQL = strSQL + " WHERE (sn_cita.fhora_cita>=" + Format$(PCurrent, "yyyy-mm-
dd") & " 00:00:00" '
    strSQL = strSQL + " And sn_cita.fhora_cita<=" + Format$(PCurrent, "yyyy-mm-dd")
& " 23:59:59'"
Set rsCita = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
If Not (rsCita.EOF And rsCita.BOF) Then
    While Not rsCita.EOF
        Sn = Trim(rsCita(0))
        citas = 0
        For H = 1 To introwg
            If FBkListAmp.TextRC(H, 4) = Sn Then
                citas = 1
                Exit For
            End If
        Next
        If citas = 0 Then
            FBkListAmp.MaxRow = introwg + 1
            introwg = introwg + 1
            Cita3 (Sn)
        Else
            FBkListAmp.MaxRow = introwg
        End If
        rsCita.MoveNext
    Wend
End If
FBkListAmp.MaxRow = introwg
SQL = "SELECT sn_asigna_ajust.no_siniestro "
SQL = SQL + "From sn_asigna_ajust sn_asigna_ajust "
SQL = SQL + "WHERE (sn_asigna_ajust.env_mensaje='S')'"
Set rs = Cnet.OpenResultset(SQL, 0, rdConcurReadOnly)
If rs.EOF And rs.BOF Then
    'no hay siniestros
Else
    While Not rs.EOF
        Sn = Trim(rs(0))
        For H = 1 To introwg
            If FBkListAmp.TextRC(H, 4) = Sn Then
                FBkListAmp.TextRC(H, 1) = "@"
            End If
        Next
        rs.MoveNext
    Wend
End If
Tiempo_IA
Else
    FBkListAmp.MaxRow = 1
    MsgBox "No se registraron siniestros", vbExclamation
End If
End Sub

```

Public Function Ajustadores(gRow1 As String, tipo_ajus As String)

```

** PROPOSITO      : Asignación ajustador de auto
**
** PARAMETROS    : grow1 as string, Tipo_Ajus as Integer
**
** REGRESO       :
**
Dim noempleado As Long
Dim Aviso As String
Dim Llegada As String
Dim Termino As String
Dim Retiro As String

```

```

'Query para verificar que existe el número de siniestro
strSQL1 = "Select sn_asigna_ajust.no_empleado, sn_asigna_ajust.fec_aviso,"

```

```

    strSQL1 = strSQL1 + "sn_asigna_ajust.fec_llegada, sn_asigna_ajust.fec_termino,
sn_asigna_ajust.fec_retiro_aseg"
    strSQL1 = strSQL1 + " From sn_asigna_ajust sn_asigna_ajust, sn_siniestro sn_siniestro"
    strSQL1 = strSQL1 + " Where sn_asigna_ajust.no_siniestro = 'I' & gRow1 & "I"
    strSQL1 = strSQL1 + " AND sn_siniestro.no_siniestro = sn_asigna_ajust.no_siniestro"
    strSQL1 = strSQL1 + " AND ((sn_asigna_ajust.cve_tipo_ajus= " & tipo_ajus & ")
    strSQL1 = strSQL1 + " AND (sn_asigna_ajust.cve_est_aten <> 26003))"
Set rs = Cnet.OpenResultset(strSQL1, 0, rdConcurReadOnly)

If rs.EOF And rs.BOF Then
    If tipo_ajus = 24002 Then
        Asig_Ajustador tipo_ajus, CboAjusAuto
        K = K + 1
    End If
    If tipo_ajus = 24001 Then
        Asig_Ajustador tipo_ajus, CboAjusMoto
        K = K + 1
    End If
    If tipo_ajus = 24003 Then
        Asig_Ajustador tipo_ajus, CmbDependencia
        CmbDependencia.Enabled = True
        K = K + 1
    End If
Else
    If tipo_ajus = 24002 Then
        noempleado = rs(0)
        Aviso = ChecarNulls(rs(1))
        Llegada = ChecarNulls(rs(2))
        Termino = ChecarNulls(rs(3))
        Retiro = ChecarNulls(rs(4))
        Empleado noempleado, CboAjusAuto
        CboAjusAuto.Enabled = False
        CmdAvisoA(0).Enabled = False
        If Aviso <> "" Then
            MaskedTextBox2(8) = Format$(CDate(Aviso), "hh:mm:ss")
        End If
        CmdArribaA(0).Enabled = True
        If Llegada <> "" Then
            CmdArribaA(0).Enabled = False
            MaskedTextBox2(4) = Format$(CDate(Llegada), "hh:mm:ss")
            CmdRetiroA(0).Enabled = True
            CmdTerminoA(0).Enabled = True
        End If
        If Retiro <> "" Then
            CmdRetiroA(0).Enabled = False
            MaskedTextBox2(5) = Format$(CDate(Retiro), "hh:mm:ss")
            CmdTerminoA(0).Enabled = True
            CmdRetiroM(1).Enabled = False
            MaskedTextBox2(10).Text = MaskedTextBox2(5)
        End If
        If Termino <> "" Then
            CmdTerminoA(0).Enabled = False
            MaskedTextBox2(6) = Format$(CDate(Termino), "hh:mm:ss")
            FrmAjusAuto.Enabled = False
        End If
    End If
    If tipo_ajus = 24001 Then
        noempleado = rs(0)
        Aviso = ChecarNulls(rs(1))
        Llegada = ChecarNulls(rs(2))
        Termino = ChecarNulls(rs(3))
        Retiro = ChecarNulls(rs(4))
        Empleado noempleado, CboAjusMoto
        CmdAvisoM(1).Enabled = False
        CboAjusMoto.Enabled = False
        If Aviso <> "" Then
            MaskedTextBox2(7) = Format$(CDate(Aviso), "hh:mm:ss")
            CmdArribaM(1).Enabled = True
        End If
        If Llegada <> "" Then

```

```

        CmdArriboM(1).Enabled = False
        MaskEdBox2(9) = Format$(CDate(Llegada), "hh:mm:ss")
        CmdRetiroM(1).Enabled = True
        CmdTerminoM(1).Enabled = True
    End If
    If Retiro <> "" Then
        CmdRetiroM(1).Enabled = False
        MaskEdBox2(10) = Format$(CDate(Retiro), "hh:mm:ss")
        CmdTerminoM(1).Enabled = True
        MaskEdBox2(5) = MaskEdBox2(10)
        CmdRetiroA(0).Enabled = False
    End If
    If MaskEdBox2(5).Text <> "_: :_" Then
        CmdRetiroM(1).Enabled = False
    End If
    If Termino <> "" Then
        CmdTerminoM(1).Enabled = False
        MaskEdBox2(11) = Format$(CDate(Termino), "hh:mm:ss")
        FrmAjMoto.Enabled = False
    End If
End If

If tipo_ajus = 24003 Then
    noempleado = rs(0)
    Empleado noempleado, CmbDependencia
    CmbDependencia.Enabled = False
End If

End If

End Function

Public Sub GridAjusta()
    '* PROPOSITO      : Checa el número de Ajustador para colocarlo en el Grid
    '*
    '* PARAMETROS      :
    '*
    '* REGRESO         :
    '*
    Dim sini() As String
    Dim ajus() As String
    Dim sini1() As String
    Dim ajus1() As String
    ReDim txtsini1(2001) As String
    Colum = 4
    p = 1
    qRow = FBKListAmp.Row
    %Siniestro = FBKListAmp.TextRC(qRow, Colum)
    nrow = FBKListAmp.MaxRow

    strSQL = "SELECT sn_asigna_ajust.no_siniestro, sn_directorio_mul.cve_ajustador "
    strSQL = strSQL + " From sn_asigna_ajust sn_asigna_ajust, sn_directorio_mul
sn_directorio_mul "
    strSQL = strSQL + " Where sn_asigna_ajust.No_Empleado = sn_directorio_mul.cve_usuario
"
    strSQL = strSQL + " And ((sn_asigna_ajust.cve_tipo_ajus = 24002) "
    strSQL = strSQL + " And (sn_asigna_ajust.cve_est_ajen <> 26003) "
    strSQL = strSQL + " And (sn_directorio_mul.cve_cv = " & CveCV & "))"
    Set rs = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)

    L = 1
    ReDim sini(100) As String
    ReDimaju(100) As String
    While Not rs.EOF
        sini(L) = ChecarNulls(rs(0))
        ajus(L) = ChecarNulls(rs(1))
        rs.MoveNext
        L = L + 1
    Wend

```

```

Do While p <= mrow
    txtsinies(p) = FBkListAmp.TextRC(p, Colum)
    If txtsinies(p) <> "" Then
        For Q = 1 To L - 1
            If txtsinies(p) = Trim(sini(Q)) And txtsinies(p) <> "" Then
                FBkListAmp.TextRC(p, 6) =aju(Q)
            ElseIf txtsinies(p) = "" Then
                Exit For
            End If
        Next
        p = p + 1
    Else
        Exit Do
    End If
Loop

' ***** Para los ajustadores de moto *****
 strSQL = "SELECT sn_asigna_ajust.no_siniestro, sn_directorio_mul.cve_ajustador "
 strSQL = strSQL + " From sn_asigna_ajust sn_asigna_ajust, sn_directorio_mul
sn_directorio_mul "
 strSQL = strSQL + " Where sn_asigna_ajust.No_Empleado = sn_directorio_mul.cve_usuario
"
 strSQL = strSQL + " And ((sn_asigna_ajust.cve_tipo_ajus = 24001) "
 strSQL = strSQL + " And (sn_asigna_ajust.cve_est_aten <> 26003) "
 strSQL = strSQL + " And (sn_directorio_mul.cve_cv= " & CveCV & ")))"
 Set rs = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
 L = 1
 p = 1
 ReDim sinil(100) As String
 ReDim ajul(100) As String
 While Not rs.EOF
    sinil(L) = rs(0)
    ajul(L) = rs(1)
    rs.MoveNext
    L = L + 1
 Wend

Do While p <= mrow
    txtsinies(p) = FBkListAmp.TextRC(p, Colum)
    If txtsinies(p) <> "" Then
        For Q = 1 To L
            If txtsinies(p) = Trim(sinil(Q)) And txtsinies(p) <> "" Then
                FBkListAmp.TextRC(p, 7) = .ajul(Q)
            ElseIf txtsinies(p) = "" Then
                FBkListAmp.TextRC(p, 7) = ""
            Exit For
            End If
        Next
        p = p + 1
    Else
        Exit Do
    End If
Loop

End Sub

Public Sub Asig_Ajustador(tipo_ajus As String, combo As ComboBox)
'* PROPOSITO      : Asignación ajustador de acuerdo al tipo de ajustador
'*
'* PARAMETROS    : tipo_ajus as string, combo as ComboBox
'*
'* REGRESO       :
'*
Dim ajustadl() As String
Dim ajt As String
Dim kk As Boolean
Dim Nombre As String
Dim Paterne As String

```

```

Dim Materno As String
Dim blnBan As Boolean

blnBan = False
' **** Tipo de Siniestro ****
strSQL = "Select A.cve_tsiniestro from sn_siniestro A Where no_siniestro=" &
NoSiniestro & "1"
Set rs1 = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
' **** Ajustadores ****
strSQL = " SELECT DM.cve_ajustador, DM.nombre, DM.a_paterno, DM.a_materno "
strSQL = strSQL + " FROM sn_asigna_turno SAT, sn_directorio_mul DM, sn_zona_base ZB "
strSQL = strSQL + " WHERE SAT.no_employado = DM.cve_usuario "
strSQL = strSQL + " AND SAT.cve_base = ZB.cve_base "
strSQL = strSQL + " AND ((ZB.cve_cv=1) "
strSQL = strSQL + " AND (SAT.dia_semana=" + Format$(PCurrent, "w") + ") "
strSQL = strSQL + " AND (SAT.cve_tipo_ajus=" & tipo_ajus & ") "
strSQL = strSQL + " AND (DM.habilitado = 'S')"
If ChecarNullsNum(rs1(0)) = 10 Then
    strSQL = strSQL + " AND ZB.cve_base=80 "
End If
strSQL = strSQL + " AND SAT.fec_inicio <= current "
strSQL = strSQL + " AND current <= SAT.fec_fin "
strSQL = strSQL + " AND " & Format$(PCurrent, "hh:mm") & " " between SAT.hora_inicial
AND SAT.hora_final "
Set rs = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
ReDim Ajustad(100) As String
JI = 1
If rs.EOF And rs.BOF Then
    Ajus_Otros tipo_ajus, combo
    MousePointer = 0
    Exit Sub
    On Error GoTo 0
Else
    combo.Style = 2
    combo.Enabled = True
    If tipo_ajus = 24003 Then
        While Not rs.EOF
            combo.AddItem ChecarNulls(rs(0)) + " - " + ChecarNulls(rs(1)) + " " +
ChecarNulls(rs(2)) + " " + ChecarNulls(rs(3))
            rs.MoveNext
        Wend
        Exit Sub
    Else
        While Not rs.EOF
            Ajustad(JI) = rs(0) + " - " + ChecarNulls(rs(1)) + " " + ChecarNulls(rs(2))
+ " " + ChecarNulls(rs(3))
            rs.MoveNext
            JI = JI + 1
        Wend
        End If
    End If

    strSQL = "SELECT DISTINCT sn_directorio_mul.cve_ajustador "
    strSQL = strSQL + "From sn_asigna_ajust sn_asigna_ajust, sn_directorio_mul
sn_directorio_mul "
    strSQL = strSQL + "Where sn_asigna_ajust.No_Employado = sn_directorio_mul.cve_usuario
"
    strSQL = strSQL + "And ((sn_asigna_ajust.cve_tipo_ajus = " & tipo_ajus & ") "
    strSQL = strSQL + "And (sn_asigna_ajust.cve_est_aten = 26001) "
    strSQL = strSQL + "And (sn_directorio_mul.cve_cv= " & CveCV & ") "
    Set rs = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
    ReDim ajustad(100) As String
    JK = 1
    If Not (rs.EOF And rs.BOF) Then
        While Not rs.EOF
            ajustad(JK) = rs(0)
            JK = JK + 1
            rs.MoveNext
        Wend
    End If
    combo.Style = 2

```

```

For IK = 1 To JI - 1
  For IJ = 1 To JK - 1
    cosa = Mid(Ajustad(IK), 1, 5)
    If cosa = ajustad1(IJ) Then
      kk = True
      Exit For
    Else
      kk = False
    End If
  Next IJ
Next IK

If kk = False Then
  combo.AddItem Ajustad(IK)      'Llena el combo seleccionado
  binBan = True
End If
If binBan = False Then
  Ajus_Otros tipo ajus, combo
  MousePointer = 0
  Exit Sub
End If
Next
MousePointer = 0

End Sub

```

```

Public Function Empleado(noempl As Long, combo As ComboBox)
  '* PROPOSITO      : Llena el combo de los ajustadores de auto o moto
  '*
  '* PARAMETROS     : noempl As Integer, combo As ComboBox
  '*
  '* REGRESO        :
  '*
  Dim Nombre As String
  Dim Paterno As String
  Dim Materno As String

  strSQL = "SELECT sn_directorio_mul.cve_ajustador, sn_directorio_mul.nombre,
sn_directorio_mul.a_paterno, sn_directorio_mul.a_materno"
  strSQL = strSQL + " FROM sn_directorio_mul sn_directorio_mul"
  strSQL = strSQL + " WHERE sn_directorio_mul.cve_usuario = " & noempl
  strSQL = strSQL + " And (sn_directorio_mul.cve_cv= " & CveCV & ")"
  Set rs = Cnet.OpenResultSet(strSQL, 0, rdConcurReadOnly)
  If IsNull(rs(1)) Then
    Nombre = ""
  Else
    Nombre = rs(1)
  End If
  If IsNull(rs(2)) Then
    Paterno = ""
  Else
    Paterno = rs(2)
  End If
  If IsNull(rs(3)) Then
    Materno = ""
  Else
    Materno = rs(3)
  End If
  '*
  '* combo.Style = 0
  '* combo.Text = rs(0) & "- " & Nombre & " " & Paterno & " " & Materno

End Function

```

```

Public Sub Cita()
  '* PROPOSITO      : Coloca la Palabra CITA en el Grid para los casos que tenga cita
  '*
  '* PARAMETROS     :
  '*
  '* REGRESO        :
  '*

```

```

Dim sinies() As String
ReDim txtsinies(200) As String
I = 1
Colum = 4
gRow = FBkListAmp.Row
NoSiniestro = FBkListAmp.TextRC(gRow, Colum)

mrow = FBkListAmp.MaxRow
Do While I <= mrow
    txtsinies(I) = FBkListAmp.TextRC(I, Colum)
    nos = Cita2(txtsinies(I))
    If txtsinies(I) = Trim(nositro) And txtsinies(I) <> "" Then
        FBkListAmp.TextRC(I, 2) = "CITA"
    Else
        FBkListAmp.TextRC(I, 2) = ""
    End If
    I = I + 1
Loop

End Sub

Public Sub GuardaHora(FechaHora As String, boton As String, noempleado As Long)
    '* PROPOSITO      : Guarda la Fecha y Hora en la tabla sn_asig_ajus
    '*
    '* PARAMETROS     : FechaHora As String, boton As String, noempleado As Integer
    '*
    '* REGRESO        :
    If boton <> "fec_termino" Then
        strSQL = "SELECT sn_asigna_ajust.no_siniestro "
        strSQL = strSQL + "FROM sn_asigna_ajust sn_asigna_ajust "
        strSQL = strSQL + "WHERE (sn_asigna_ajust.no_siniestro=" & NoSiniestro & " ) "
        strSQL = strSQL + "AND (sn_asigna_ajust.no_empleado=" & noempleado & " ) "
        strSQL = strSQL + "AND (sn_asigna_ajust." & boton & " Is Null)"
        Set rs = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
        If Not (rs.EOF And rs.BOF) Then
            strSQL = "UPDATE sn_asigna_ajust "
            strSQL = strSQL + " SET " & boton & " =current, fec_actualiza = current WHERE "
            strSQL = strSQL + " sn_asigna_ajust.no_siniestro=" & NoSiniestro & " "
            strSQL = strSQL + " AND sn_asigna_ajust.no_empleado=" & noempleado
            Cnet.Execute strSQL
        Else
            MsgBox "Ya ha sido guardado el tiempo en otra maquina"
            Ajustadores NoSiniestro, 24002 'Llama al procedimiento Ajustadores para
            asignar ajustador
            Exit Sub
        End If
    Else
        If DocAjusta = "24002" Then
            strSQL = strSQL + "SELECT sn_asigna_ajust.no_siniestro "
            strSQL = strSQL + "FROM sn_asigna_ajust sn_asigna_ajust "
            strSQL = strSQL + "WHERE (sn_asigna_ajust.no_siniestro=" & NoSiniestro & " ) "
            strSQL = strSQL + "AND (sn_asigna_ajust.no_empleado=" & noempleado & " ) "
            strSQL = strSQL + "AND (sn_asigna_ajust." & boton & " Is Null)"
            Set rs = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
            If Not (rs.EOF And rs.BOF) Then
                strSQL = "UPDATE sn_asigna_ajust "
                strSQL = strSQL + " SET " & boton & " =-current, fec_actualiza = current,"
                If CboAjustMoto.Text <> "" And MaskEdBox2(11).Text = " : : " Then
                    strSQL = strSQL + " cve_est_aten = 26002 WHERE sn_asigna_ajust.no_siniestro
                = " & NoSiniestro & " "
                Else
                    strSQL = strSQL + " cve_est_aten = 26003 WHERE sn_asigna_ajust.no_siniestro
                = " & NoSiniestro & " "
                End If
                strSQL = strSQL + " AND sn_asigna_ajust.no_empleado = " & noempleado
                Cnet.Execute strSQL
            Else
                MsgBox "Ya ha sido guardado el tiempo en otra maquina"
                Ajustadores NoSiniestro, 24002 'Llama al procedimiento Ajustadores para
                asignar ajustador
                Exit Sub
            End If
        End If
    End If
End Sub

```

```

End If
End If
If DocAjusta = "24001" Then
  strSQL = strSQL + "SELECT sn_asigna_ajust.no_siniestro "
  strSQL = strSQL + "From sn_asigna_ajust sn_asigna_ajust "
  strSQL = strSQL + "WHERE (sn_asigna_ajust.no_siniestro=" & NoSiniestro & ") "
  strSQL = strSQL + "AND (sn_asigna_ajust.no_empleado=" & noempleado & ") "
  strSQL = strSQL + "AND (sn_asigna_ajust." & boton & " Is Null)"
  Set rs = Cnet.OpenResultset(strSQL, 0, rdConcurReadOnly)
  If Not (rs.EOF And rs.BOF) Then
    strSQL = "UPDATE sn_asigna_ajust "
    strSQL = strSQL + " SET " & boton & "=current , fec_actualiza = current , "
    If CboAjustaAuto.Text <> "" And MaskEdBox2(6).Text = "_:_" Then
      strSQL = strSQL + " cve_est_aten = 26002 WHERE sn_asigna_ajust.no_siniestro="
      strSQL & NoSiniestro & " "
    Else
      strSQL = strSQL + " cve_est_aten = 26003 WHERE sn_asigna_ajust.no_siniestro="
      strSQL & NoSiniestro & " "
    End If
    strSQL = strSQL + " AND sn_asigna_ajust.no_empleado=" & noempleado
    Cnet.Execute strSQL
  Else
    MsgBox "Ya ha sido guardado el tiempo en otra maquina"
    Ajustadores NoSiniestro, 24001
    Exit Sub
  End If
End If
End If
End Sub

```

End Sub

Public Sub Dependencia(Siniestro1 As String)

```

** PROPOSITO      : Llena los campos del siniestro dado para la asignación de
ajustador de dependencia*
**
** PARAMETROS    : Siniestro1 as String
**
** REGRESO      :
**
' SELECT DISTINCT sn_siniestro.fec_siniestro, sn_sini_poliza.poliza,
sn_sini_poliza.inciso, _
sn_aseguradora.nom_aseguradora, sn_siniestro.lugar
' From sn_aseguradora sn_aseguradora, sn_sini_poliza sn_sini_poliza, sn_sinies_veh1
sn_sinies_veh1, _
sn_siniestro sn_siniestro, sn_zona_base sn_zona_base
' WHERE sn_sini_poliza.cve_aseguradora = sn_aseguradora.cve_aseguradora _
AND sn_sinies_veh1.no_siniestro = sn_sini_poliza.no_siniestro _
AND sn_sinies_veh1.cve_base_sini = sn_zona_base.cve_base _
AND sn_siniestro.no_siniestro = sn_sinies_veh1.no_siniestro _
AND ((sn_siniestro.no_siniestro='13274') AND (sn_zona_base.es_dependencia='D'))

strSQL = "SELECT DISTINCT sn_siniestro.fec_siniestro, sn_sini_poliza.poliza, "
strSQL = strSQL + "sn_sini_poliza.inciso, sn_aseguradora.nom_aseguradora, "
sn_siniestro.lugar, "
strSQL = strSQL + "sn_zona_base.cve_base, sn_zona_base.descripcion "
strSQL = strSQL + "From sn_aseguradora sn_aseguradora, sn_sini_poliza sn_sini_poliza,
sn_sinies_veh1 sn_sinies_veh1, "
strSQL = strSQL + "sn_siniestro sn_siniestro, sn_zona_base sn_zona_base "
strSQL = strSQL + "Where sn_sini_poliza.cve_aseguradora = "
sn_aseguradora.cve_aseguradora "
strSQL = strSQL + "AND sn_sinies_veh1.no_siniestro = sn_sini_poliza.no_siniestro "
strSQL = strSQL + "AND sn_sinies_veh1.cve_base_sini = sn_zona_base.cve_base "
strSQL = strSQL + "AND sn_siniestro.no_siniestro = sn_sinies_veh1.no_siniestro "
strSQL = strSQL + "AND ((sn_siniestro.no_siniestro=" & Siniestro1 & ") AND
(sn_zona_base.es_dependencia='D'))"
strSQL = strSQL + " And (sn_siniestro.cve_cv= " & CveCV & " )"
Set rs = Cnet.OpenResultset(strSQL, rdOpenKeyset, rdConcurRowVer)
If rs.EOF And rs.BOF Then
  MsgBox "No existen datos o No es Siniestro de Dependencia"
  TxtSini.Text = ""

```

```

Else
  Txt18(1).Text = Siniestrol
  dato = ChecarNulls(rs(0))
  Colocar_MskFecha MaskEdBox3(1), dato
  Separa_FyH dato, MskHrsSini(3)
  Txt19(1).Text = ChecarNulls(rs(1))
  Txt20(1).Text = ChecarNulls(rs(2))
  Txt21(0).Text = ChecarNulls(rs(3))
  Txt22(1).Text = ChecarNulls(rs(4))
  CmbAjustador(1).Text = ChecarNulls(rs(5)) & " - " & ChecarNulls(rs(6))
  Frajusdep.Visible = True
End If
End Sub

```

Public Sub Mensaje_Ajus(no_sinie As String, TAjus As Integer)

```

' * PROPOSITO           : Construye el mensaje para pasa los datos al componente
' *
' * PARAMETROS         : no_sinie
' *
' * REGRESO            : TAjus
' *
  strSQL = "SELECT sn_aseguradora.nom_aseguradora, sn_siniestro.no_siniestro,
sn_poliza.poliza, sn_poliza.inciso, sn_poliza.proce_x_Cobranza,"
  strSQL = strSQL + "sn_poliza.fec_invigencia, sn_poliza.fec_finvigencia,
sn_poliza.nom_asegurado||' ||sn_poliza.apat_asegurado||' ||sn_poliza.amat_asegurado, "
  strSQL = strSQL + "sn_sinies_vehl.nom_conduc||' ||sn_sinies_vehl.apat_conduc||'
' ||sn_sinies_vehl.amat_conduc, sn_sinies_vehl.tel_casa_conduc,"
  strSQL = strSQL + "sn_sinies_vehl.tel_ofi_conduc, sn_siniestro.tec_siniestro,
sn_tipo_sini.cve_tsiniestro, sn_tipo_sini.descripcion, "
  strSQL = strSQL + "sn_zona_base.cve_base, sn_zona_base.descripcion,
sn_siniestro.lesionados, sn_sinies_vehl.transito, "
  strSQL = strSQL + "sn_cat.descripcion, sn_poliza.color, sn_poliza.modelo,
sn_poliza.placas, sn_poliza.no_serie, sn_poliza.cve_segnty, "
  strSQL = strSQL + "sn_sinies_vehl.valor_comercial, sn_siniestro.lugar "
  strSQL = strSQL + "From sn_aseguradora sn_aseguradora, sn_cat sn_cat, sn_poliza
sn_poliza, sn_siniestro sn_siniestro, "
  strSQL = strSQL + "sn_sinies_vehl sn_sinies_vehl, sn_siniestro sn_siniestro,
sn_tipo_sini sn_tipo_sini, sn_zona_base sn_zona_base "
  strSQL = strSQL + "WHERE sn_poliza.cve_marca = sn_cat.cve_catalogo "
  strSQL = strSQL + "AND sn_poliza.inciso = sn_siniestro.inciso "
  strSQL = strSQL + "AND sn_poliza.poliza = sn_siniestro.poliza "
  strSQL = strSQL + "AND sn_siniestro.no_siniestro = sn_sinies_vehl.no_siniestro "
  strSQL = strSQL + "AND sn_siniestro.cve_tsiniestro = sn_tipo_sini.cve_tsiniestro "
  strSQL = strSQL + "AND sn_sinies_vehl.no_siniestro = sn_siniestro.no_siniestro "
  strSQL = strSQL + "AND sn_sinies_vehl.cve_base_sini = sn_zona_base.cve_base "
  strSQL = strSQL + "AND sn_siniestro.cve_aseguradora = sn_aseguradora.cve_aseguradora "
  strSQL = strSQL + "AND ((sn_siniestro.no_siniestro = " & no_sinie & " )"
  Set rs = Cnet.OpenResultSet(strSQL, rdOpenKeyset, rdConcurRowVer)
  FrmMensaje.Txt1.Text = ChecarNulls(rs(0))
  FrmMensaje.Txt7.Text = ChecarNulls(rs(1))
  FrmMensaje.Txt2.Text = ChecarNulls(rs(2))
  FrmMensaje.Txt3.Text = ChecarNulls(rs(3))
  If ChecarNulls(rs(4)) = "S" Then
    FrmMensaje.Txt4.Text = "COBRANZA O. K."
  ElseIf ChecarNulls(rs(4)) = "C" Then
    FrmMensaje.Txt4.Text = "COBRANZA CANCELADA"
  ElseIf IsNull(rs(4)) Then
    FrmMensaje.Txt4.Text = "NO HAY INFORMACIÓN"
  Else
    FrmMensaje.Txt1.Text = "PEDIR DOCUMENTOS"
  End If
  FrmMensaje.Txt6.Text = ChecarNulls(rs(5))
  FrmMensaje.Txt11.Text = ChecarNulls(rs(6))
  FrmMensaje.Txt8.Text = ChecarNulls(rs(7))
  FrmMensaje.Txt9.Text = ChecarNulls(rs(8))
  FrmMensaje.Txt10.Text = ChecarNulls(rs(9))
  FrmMensaje.Txt31.Text = ChecarNulls(rs(10))
  FrmMensaje.MaskEdBox1(0).Text = Format$(ChecarNulls(rs(11)), "dd/mm/yyyy")
  Separa_FyH ChecarNulls(rs(11)), FrmMensaje.MaskEdBox2(0)

```

```

FrmMensaje.Txt13.Text = ChecarNulls(rs(12)) & " - " & ChecarNulls(rs(13))
FrmMensaje.Txt14.Text = ChecarNulls(rs(14)) & " - " & ChecarNulls(rs(15))
FrmMensaje.Txt15.Text = ChecarNulls(rs(16))
FrmMensaje.Txt16.Text = ChecarNulls(rs(17))
FrmMensaje.Txt17.Text = ChecarNulls(rs(18))
FrmMensaje.Txt18.Text = ChecarNulls(rs(19))
FrmMensaje.Txt19.Text = ChecarNulls(rs(20))
FrmMensaje.Txt20.Text = ChecarNulls(rs(21))
FrmMensaje.Txt21.Text = ChecarNulls(rs(22))
FrmMensaje.Txt22.Text = ChecarNulls(rs(23))
FrmMensaje.Txt23.Text = ChecarNulls(rs(24))
FrmMensaje.Txt24.Text = ChecarNulls(rs(25))

End Sub

Public Function PCurrent() As String
'* PROPOSITO      : Regresa la fecha y hora del sistema de informix
'*
'* PARAMETROS    :
'*
'* REGRESO       : Pcurrent as String
'*
SQL = "select current from sn_divisa"
Set rs = Cnet.OpenResultset(SQL, rdOpenKeyset, rdConcurRowVer)
If Not (rs.EOF And rs.BOF) Then
    PCurrent = ChecarNulls(rs(0))
End If
End Function

Private Sub CmdCancelaSini_Click()
'* PROPOSITO      : Cancela los Siniestros en Atención
'*
'* PARAMETROS    :
'*
'* REGRESO       :
'*
TmrListAmp.Enabled = False
txtAjusAuto = CboAjusAuto.Text
txtAjusAuto = Mid(txtAjusAuto, 1, 3)
txtAjusMoto = CboAjusMoto.Text
txtAjusMoto = Mid(txtAjusMoto, 1, 3)
FrmLibera.Caption = "GSTR_SLS_SISTEMA" & " - Cancelación del Siniestro"
FrmLibera.TxtCausaCancela.Visible = True
FrmLibera.TxtCausas.Caption = "Causas de Cancelación"
FrmLibera.Show

End Sub

Private Sub CmdVisitas_Click()
'* PROPOSITO      : Despliega pantalla con información del asegurado y de la cita
'*
'* PARAMETROS    :
'*
'* REGRESO       :
'*
Dim Nombre As String
Dim Paterno As String
Dim Materno As String
Dim Lugar As String

TmrListAmp.Enabled = False
strSQL = "SELECT sn_cita.nombre, sn_cita.a_paterno, sn_cita.a_materno,
sn_cita.hora_cita, sn_cita.telefono_casa, "
strSQL = strSQL + "sn_cita.telefono_ofi, sn_cita.lugar"
strSQL = strSQL + " FROM sn_cita sn_cita "
strSQL = strSQL + " WHERE (sn_cita.no_siniestro=" & NoSiniestro & ")"

```

```

strSQL = strSQL + " ORDER BY sn_cita.fhora_cita DESC"
Set rs = Cnet.OpenResultset(strSQL, rdOpenKeyset, rdConcurRowVer)
If rs.EOF And rs.BOF Then
    MsgBox "No hay información para esta visita"
Else
    SSTabVisita.Visible = True
    SSTabVisita.ZOrder
    If IsNull(rs(0)) Then
        Nombre = ""
    Else
        Nombre = rs(0)
    End If
    If IsNull(rs(1)) Then
        Paterno = ""
    Else
        Paterno = rs(1)
    End If
    If IsNull(rs(2)) Then
        Materno = ""
    Else
        Materno = rs(2)
    End If
    TxtVisNom.Text = Nombre + " " + Paterno + " " + Materno
    If IsNull(rs(3)) Then
        MaskEdBox1 = " / / "
        MaskEdBox1 = "00:00:00"
    Else
        MaskEdBox1 = Format$(CDate(rs(3)), "dd/mm/yyyy")
        MaskEdBox4 = Format$(CDate(rs(3)), "hh:mm:ss")
    End If
    If IsNull(rs(4)) Then
        TxtTelCasa = ""
    Else
        TxtTelCasa = rs(4)
    End If
    If IsNull(rs(5)) Then
        TxtTelOfi = ""
    Else
        TxtTelOfi = rs(5)
    End If
    If IsNull(rs(6)) Then
        TxtDiri.Text = ""
    Else
        TxtDiri.Text = rs(6)
    End If
End If

End Sub

Private Sub FBkListAmp_KeyPress(KeyAscii As Integer)
    '* PROPOSITO      : Llama el Reporte del siniestro
    '*
    '* PARAMETROS     : KeyAscii
    '*
    '* REGRESO        :
    '*
    If KeyAscii = 13 Then
        FrmListAmp.Enabled = False
        Reporte_Siniestro FBkListAmp.TextRC(FBkListAmp.Row, 1)
    End If
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
    '* PROPOSITO      : Forma General del listado Amplio, Registra la tecla ESC para
    '*                  poder tener la facilidad de salir de o cancelar cualquier
    '*                  tipo de operación
    '*
    '*
    '* PARAMETROS     : KeyAscii
    '*

```

** REGRESO

**

Dim hoy As String

If KeyAscii = 27 Then

```
If SStabVisita.Visible = True Then
    SStabVisita.Visible = False
    FBKListAmp.SetFocus
    TmrListAmp.Enabled = True
End If
If SStReporteSini.Visible = True Then
    SStReporteSini.Visible = False
    TmrListAmp.Enabled = True
    FBKListAmp.SetFocus
End If
```

If SStabl.Tab = 1 Then

Limpia_Corrección

ElseIf SStabl.Tab = 2 Then

CmbDependencia.Clear

CmbAjustador(1).Clear

CmbDependencia.Clear

'CmbDependencia.Text = "

'CmbDependencia.Text = ""

CmbDependencia.Enabled = True

If Frajusdep.Visible = True And Txtsini.Visible = True Then

Frajusdep.Visible = False

Lblfechini(0).Visible = False

MskFechini(0).Visible = False

Lblfechfin(1).Visible = False

MskFechFin(1).Visible = False

Lblsini.Visible = False

Fxtsini.Visible = False

FraBusca.Visible = True

OptSinie.Value = False

OptFech.Value = False

FBKAjusDep.Visible = False

Else

If Frajusdep.Visible = True And MskFechFin(1).Visible = True Then

Frajusdep.Visible = False

MskFechfin_keyPress 1, 13

Else

If FBKAjusDep.Visible = False And Frajusdep.Visible = False Then

Lblfechini(0).Visible = False

MskFechini(0).Visible = False

Lblfechfin(1).Visible = False

MskFechFin(1).Visible = False

Lblsini.Visible = False

Fxtsini.Visible = False

FraBusca.Visible = True

OptSinie.Value = False

OptFech.Value = False

Else

If FBKAjusDep.Visible = True Then

Frajusdep.Visible = False

Lblfechini(0).Visible = False

MskFechini(0).Visible = False

Lblfechfin(1).Visible = False

MskFechFin(1).Visible = False

Lblsini.Visible = False

Txtsini.Visible = False

FraBusca.Visible = True

OptSinie.Value = False

OptFech.Value = False

FBKAjusDep.Visible = False

End If

End If

End If

End If

ElseIf SStabl.Tab = 3 Then

```

    hoy = PCurrent
    hoy = Mid(hoy, 1, 10)
    FBkCambios.Visible = False
    'FBkCambios.MaxRow = 1
    MskFechIni_C(1).Mask = ""
    MskFechIni_C(1).Mask = "###/##/####"
    MskFechIni_C(1) = hoy
    MskFechFin_C(0).Mask = ""
    MskFechFin_C(0).Mask = "###/##/####"
    MskFechFin_C(0) = hoy
    MskFechIni_C(1).SetFocus
End If

If SStab2.Tab = 1 Then
    If FBkReactiva.Visible = True Then
        FBkReactiva.Visible = False
        hoy = PCurrent
        hoy = Mid(hoy, 1, 10)
        MskFechIni1(0).Mask = ""
        MskFechIni1(0).Mask = "###/##/####"
        MskFechIni1(0) = hoy
        MskFechFin1(1).Mask = ""
        MskFechFin1(1).Mask = "###/##/####"
        MskFechFin1(1) = hoy
        FBkReactiva.MaxCol = 1
        FBkReactiva.MaxRow = 1
        FBkReactiva.MaxCol = 13
        FBkReactiva.MaxRow = 22
        MskFechIni1(0).SetFocus
    Else
        If FrmDescSini.Visible = True Then
            FrmDescSini.Visible = False
            MskFechFin1_KeyPress 1, 13
        End If
    End If
End If

End If

End Sub

Private Sub Form_Load()
    '* PROPOSITO      : Forma General del listado Amplio, Registra la tecla ESC para *
    '*                : poder tener la facilidad de salir o de cancelar cualquier *
    '*                : tipo de operación *
    '* *
    '* PARAMETROS      : *
    '* *
    '* REGRESO         : *
    '* *
    Dim lngInst As Long
    I = 1
    K = 1
    Z = 1

    Set udtConecta = CreateObject("Conexión.UtilityClass")
    Set KeyAdmin = CreateObject("KeyClientRDO.KeyClientRDO.1")

    '* ***** Control del KeyAdmin *****
    If KeyAdmin.DlgLogin("ASISTE", "", 0) = 1 Then
        If KeyAdmin.Validate(106000) = k_ok Then
            If KeyAdmin.Validate(100000) = k_ok Then
                If udtConecta.Conectar(Conet, KeyAdmin.sDBName, KeyAdmin.sDBUser,
                    KeyAdmin.sDBPwd) Then
                    MsgBox "Conexión lograda"
                    FrmRadio.Caption = GSTR_SLS_SISTEMA + " - R A D I O"
                Else
                    MsgBox "Por el momento no hay conexión a la BD"
                End
            End If
        End If
    End If

```

```

        End If
    Else
        MsgBox "No tienes acceso a este sistema"
    End
End If
Else
    MsgBox " no tienes acceso a este módulo"
End If
Else
    MsgBox "No tienes acceso"
End
End If

' ***** Inicia el Modulo de Radio *****

If App.PreviousInstance Then
    MsgBox "Ya se tiene una aplicación abierta ", vbExclamation
    Unload FrmKradio
End If
Conecta 'Realiza el Query del Listado Ampio
Cita 'Checa las citas por Sinistro
GridAjusta 'coloca en el grid los ajustadores de cada siniestro cn sus horas
Tiempo_Arriba
jRow1 = FBKListAmp.Row
NoSinistro = ""
FrmKradio.ZOrder
TmrListAmp.Enabled = True

End Sub

```

Private Sub Form_Unload(Cancel As Integer)

```

' * PROPOSITO      : Libera la Forma y el evento de la memoria
' *
' * PARAMETROS    : Cancel
' *
' * REGRESO       :
' *
    Chat.Close
End Sub

```

Private Sub SStabl_Click(PreviousTab As Integer)

```

' * PROPOSITO      : Tab General del Modulo de Radio
' *
' * PARAMETROS    : PreviousTab
' *
' * REGRESO       :
' *
    Dim hoy As String

    If SStabl.Tab = 0 Then
        Toolbar1.Buttons(7).Visible = True
        Toolbar1.Buttons(7).Enabled = False
        FBKListAmp.SetFocus
        StatusBar.Panels(1).Text = "Controlador de Radio"
        If SStabl.Tab = 0 Then
            TmrListAmp.Enabled = True
        End If
    ElseIf SStabl.Tab = 1 Then
        Toolbar1.Buttons(7).Visible = False
        TmrCorreAjust1.SetFocus
        Limpia_Corrección
        StatusBar.Panels(1).Text = "Corrección de Tiempos a Sinistro"
        TmrListAmp.Enabled = False
    ElseIf SStabl.Tab = 2 Then

```

```

Toolbar1.Buttons(7).Visible = False
Frajusdep.Visible = False
Lb1fechini(0).Visible = False
MskFechIni(0).Visible = False
Lb1fechfin(1).Visible = False
MskFechFin(1).Visible = False
Lb1sini.Visible = False
Txsini.Visible = False
FraBusca.Visible = True
OptSini.Value = False
OptFech.Value = False
FBkAjusDep.Visible = False
CmbDependencia.Text = ""
CmbDependencia.Text = ""
CmbDependencia.Enabled = True
StsBar.Panels(1).Text = "Asignación de Ajustadores de Dependencia"
TmrListAmp.Enabled = False
ElseIf SSTab1.Tab = 3 Then
Toolbar1.Buttons(7).Visible = False
hoy = PCurrent
hoy = Mid(hoy, 1, 10)
FRkCambios.Visible = False
MskFechIni_C(1).Mask = ""
MskFechIni_C(1).Mask = "###/##/####"
MskFechIni_C(1) = hoy
MskFechFin_C(0).Mask = ""
MskFechFin_C(0).Mask = "###/##/####"
MskFechFin_C(0) = hoy
MskFechIni_C(1).SetFocus
StsBar.Panels(1).Text = "Consulta de Cambios de Sinistros de Aseguradora"
TmrListAmp.Enabled = False
End If
End Sub

```

Private Sub SSTab2_Click(PreviousTab As Integer)

```

** PROPOSITO      :  Iao del controlador de radio que contiene el listado amplio *
**               :  y la reactivación de siniestro *
**               :  *
** PARAMETROS    :  Index *
** PARAMETROS    :  KeyAscii *
**               :  *
** RECURSO      :  *
**               :  *
Dim hoy As String

If SSTab2.Tab = 1 Then
Toolbar1.Buttons(7).Visible = False
hoy = PCurrent
hoy = Mid(hoy, 1, 10)
MskFechIni1(0).Mask = ""
MskFechIni1(0).Mask = "###/##/####"
MskFechIni1(0) = hoy
MskFechFin1(1).Mask = ""
MskFechFin1(1).Mask = "###/##/####"
MskFechFin1(1) = hoy
FRkReactiva.Visible = False
TmrDepoSini.Visible = False
MskFechIni1(0).SetFocus
FBkReactiva.MaxCol = 1
FRkReactiva.MaxRow = 1
FBkReactiva.MaxCol = 13
FBkReactiva.MaxRow = 22
StsBar.Panels(1).Text = "Reactivación de Sinistros"
TmrListAmp.Enabled = False
ElseIf SSTab2.Tab = 2 Then
Toolbar1.Buttons(7).Visible = True
StsBar.Panels(1).Text = "Controlador de Radio"
LimpiaMascara
MkSiniestro = ""
TmrListAmp.Enabled = True

```

```
End If
End Sub
```

```
Private Sub TmrListAmp_Timer()
```

```
  '* PROPOSITO      : Timer que se activa cada 5 segundos para chequear el listado amplior
```

```
  '*
```

```
  '* PARAMETROS    : Index as Integer, KeyAscii as Integer
```

```
  '*
```

```
  '* REGRESO      :
```

```
  '*
```

```
  Dim intMR As Integer
```

```
  Dim intMaxRow As Integer
```

```
  intMR = FBkListAmp.MaxRow
```

```
  For intLL = 1 To intMR
```

```
    For intMM = 1 To 11
```

```
      FBkListAmp.TextRC(intLL, intMM) = ""
```

```
    Next
```

```
  Next
```

```
  Conccta
```

```
  Cita
```

```
  GridAjusta
```

```
  Tiempo_LA
```

```
  Tiempo_Arriba
```

```
End Sub
```

BIBLIOGRAFIA

1) "Análisis Estructurado Moderno"

Yourdon, Edward

Ed. Prentice Hall Hispanoamericana, S. A.

México 1989.

2) "La Esencia de Visual Basic 4"

Heyman, Mark Steven

Ed. Pentice Hall Hispanoamericana, S. A.

México 1996.

3) "Microsoft Education and Certication"

Mastering Microsoft® Visual Basic 5 For experienced Developers

By Microsoft Corporation 1997

4) "Redes de Odenadores"

Tenenbaum, Adrew S.

Ed. Prentice Hall Hispanoamericana, S. A.

México 1991..