

T  
621.3875  
ESPC.



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**TÓPICO DE GRADUACIÓN**

**CONMUTACIÓN TELEFÓNICA Y  
TELEFONÍA CELULAR**

**Proyecto realizado previa a la obtención del Título:  
INGENIERO EN ELECTRICIDAD**

**Especialización:  
ELECTRÓNICA**

**Tema:  
ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN  
SISTEMA PARA LA DETECCIÓN, MONITOREO Y  
ENVIO A UN BEEPER DE LAS ALARMAS DE  
SEGURIDAD, INSTALADAS EN LAS BASES  
CELULARES DE LA COMPAÑÍA CONECEL-  
GUAYAQUIL UTILIZANDO TECNOLOGÍA CELULAR.**

**Realizado por:  
RICARDO L. ESPINOZA AMAGUAÑA  
MIGUEL INTRIAGO BASURTO  
XAVIER MALDONADO VILLAVICENCIO  
JORGE PESANTES SARMIENTO**

**GUAYAQUIL - ECUADOR  
1997**



Biblioteca Central



\*D-18512\*

# TRIBUNAL

Ing. Raúl Noriega  
MIEMBRO PRINCIPAL

Ing. Wasington Medina  
DIRECTOR DEL TOPICO

Dr. Freddy Villao.  
PRESIDENTE



Biblioteca Central



**Biblioteca Central**

## **DEDICATORIA**

Dedicamos a los futuros profesionales en Telecomunicaciones el siguiente proyecto, para que amplíen sus conocimientos, así como la utilidad que se puede obtener en esta nueva tecnología como lo es la Telefonía Celular.



**Biblioteca Central**

## **AGRADECIMIENTO**

Queremos primero expresar nuestros más gratos agradecimientos a **DIOS** por el valor y la FE que nos ha dado para poder lograr uno de los más grandes objetivos en nuestra vida.

A nuestros Padres por el apoyo y paciencia que nos han brindado en todo instante.

Al **ING. VICENTE SALTOS BRAGANZA**, Profesor del Tópico por todo su apoyo moral, académico y técnico que nos ha dado en el transcurso del mismo, y para el desarrollo de nuestro proyecto.

También a los Sres. **ING. VICENTE GARCIA** e **ING. YAMIL FRANCO**, así como al **Tcnlg. KLONFER VILLOTA** y queremos hacer extensivo nuestro especial agradecimiento al **ING. SEGUNDO GARCIA** por el apoyo y la colaboración incondicional que hemos recibido.



**Biblioteca Central**

**DECLARACION EXPRESA**

**“LA RESPONSABILIDAD POR LOS HECHOS, IDEAS Y DOCTRINAS EXPUESTAS EN ESTE TRABAJO, ME CORRESPONDEN EXCLUSIVAMENTE; Y, EL PATRIMONIO INTELECTUAL DE LA MISMA A LA ESCUELA SUPERIOR POLITECNICA DEL LITORAL”**

**(REGLAMENTO DE EXAMENES Y TITULOS PROFESIONALES DE LA ESPOL)**

**XAVIER MALDONADO VILLAVICENCIO  
MIGUEL INTRIAGO BASURTO  
RICARDO ESPINOZA AMAGUAÑA  
JORGE PESANTES SARMIENTO**

*[Handwritten signatures]*  
.....  
.....  
.....  
.....



**Biblioteca Central**



**Biblioteca Central**

## **TEMARIO**

### **CAPITULO I**

#### **CARACTERÍSTICAS DE LAS ALARMAS DENTRO DE LA RED CELULAR**

- 1.1 CONFIGURACIÓN DE LA RED CELULAR EN GUAYAQUIL
- 1.2 CARACTERISTICAS DE LAS ALARMAS
- 1.3 CATEGORÍA DE LAS ALARMAS
- 1.4 CLASIFICACIÓN DE LAS ALARMAS
- 1.5 ATENCIÓN Y REPARACIÓN
- 1.6 DESCRIPCIÓN DE LAS ALARMAS EXTERNAS QUE SE PRESENTAN EN UNA CELDA
- 1.7 CONECCIÓN TÍPICA DE LAS ALARMAS EN EL ACU
- 1.8 FORMA DE OPERACIÓN DE LA UNIDAD DE CONTROL Y ALARMA (ACU)

### **CAPITULO II**

#### **ESQUEMA GENERAL DEL SISTEMA**

- 2.1 DESCRIPCIÓN DE LOS EQUIPOS
- 2.2 EQUIPOS A IMPLEMENTARSE
- 2.3 DIAGRAMA DE BLOQUES

### **CAPITULO III**

#### **PROCESAMIENTO GENERAL DEL SISTEMA**

3.1 FUNCIONAMIENTO AUTOMATICO DEL SISTEMA

3.2 MANTENIMIENTO DE LA BASE DE DATOS

### **CAPÍTULO IV**

#### **IMPLEMENTACIÓN DEL PROYECTO**

4.1 HARDWARE

4.2 SOFTWARE

### **CAPITULO V**

#### **PRUEBAS Y OPERACIÓN DEL SISTEMA**

5.1 PRUEBAS DEL SISTEMA

5.2 OPERACIÓN DEL SISTEMA

5.3 EJEMPLO DE UNA PRUEBA EXITOSA



**Biblioteca Central**



**Biblioteca Central**

## **INTRODUCCIÓN**

En el mundo actual, vivimos un gran desarrollo tecnológico en las Telecomunicaciones, comenzando desde la telegrafía pasamos a la telefonía pública, al sistema satelital y en la actualidad vivimos la telefonía celular. Podríamos afirmar que la telefonía celular es el máximo desarrollo que se ha logrado, en las comunicaciones, y que en nuestro país llegó a inicios de la década de los noventa; podemos comunicarnos a cualquier parte del mundo desde el lugar donde exista cobertura.

Los equipos que intervienen en la telefonía celular poseen un sistema de alarmas propio, para asegurar un buen funcionamiento, denominadas alarmas internas; utilizando este mismo medio y la flexibilidad del sistema, nuestro proyecto presentará la implementación de un nuevo grupo de alarmas, externas al sistema, que son utilizadas como alarmas de control y seguridad para la compañía operadora del servicio.

Actualmente las alarmas de seguridad son de gran interés y utilización en nuestro medio, pues de ello depende la calidad del servicio que se preste.

Para implementar las alarmas de seguridad, utilizando la tecnología celular, nuestro estudio e investigación está desarrollado en tres grandes grupos.

- 1.- En el conocimiento técnico del tipo de alarma que se debe y se vaya a utilizar; es decir, las alarmas que sean necesarias tales como, sensores de humo y fuego, alarmas de energía, alarma de intruso, alarma de enfriamiento, etc. que son las más utilizadas en un sistema de seguridad. Una vez instaladas las alarmas, para un mejor control de observación y monitoreo, se las concentrarán en una Unidad de Control y Alarma (ACU), que es un



equipo incorporado en los sistemas de telefonía celular de la operadora donde se está implementando este diseño.

2.- El hardware, consiste en la implementación de la alarmas que van a ser sensadas a través del ACU. Las características de cada una ellas deben ser compatibles con las especificaciones técnicas del ACU para que pueda sensarlas; tales como alarmas de contacto abierto o cerrado; el nivel de voltaje que toda alarma requiere para estar energizada, con respecto al nivel de voltaje que el ACU debe mantener para que pueda sensarla.

El monitoreo de las alarmas, será desde un terminal dedicado exclusivamente a recibir los datos de las alarmas que se generan. El terminal tendrá una salida vía módem que enviará la información al beeper del supervisor, notificando la generación de una alarma.

3.- Para llevar a cabo este procedimiento, el software implementado recibirá las alarmas que se generen, una base de datos tendrá almacenada todas las alarmas de seguridad; una vez que se genere una alarma será comparada con la base de datos e indicará la presencia del mismo enviándolo por códigos al beeper del supervisor.

La importancia de crear la automatización con un sistema adecuado, surge del siguiente razonamiento; imaginemos ¿qué sucedería si de tantas alarmas internas y externas presentes en un terminal del conmutador telefónico móvil (MTX), se descartara alguna de gran importancia debido al descuido del operador de turno en dicho terminal, su consecuencia podría ser muy grave, esto podría evitarse creando un sistema adecuado que utilice la tecnología celular, y que lo desarrollaremos en este proyecto investigativo.



# INDICE

## CAPITULO I

### CARACTERÍSTICAS DE LAS ALARMAS DENTRO DE LA RED

#### CELULAR

1.1 CONFIGURACIÓN DE LA RED CELULAR EN GUAYAQUIL, .....	Pág 11
1.2 CARACTERISTICAS DE LAS ALARMAS,.....	15
1.2.1 CONCEPTO DE ALARMA,.....	15
1.2.2 ALARMAS DE SEGURIDAD,.....	15
1.2.3 UNIDAD DE CONTROL Y ALARMA (ACU),.....	15
1.3 CATEGORÍA DE LAS ALARMAS, .....	16
1.3.1 CATEGORÍA A1, .....	16
1.3.2 CATEGORÍA A2, .....	16
1.3.3 CATEGORÍA A3, .....	16
1.4 CLASIFICACIÓN DE LAS ALARMAS,.....	17
1.4.1 ALARMAS INTERNAS,.....	17
1.4.2 ALARMAS EXTERNAS,.....	17
1.4.3 ALARMAS DE OBSERVACIÓN,.....	17
1.5 ATENCIÓN Y REPARACIÓN,.....	17
1.6 DESCRIPCIÓN DE LAS ALARMAS EXTERNAS QUE SE PRESENTAN EN UNA CELDA,.....	18
1.6.1 ALARMA DE ENERGÍA, .....	18
1.6.2 ALARMA SENSOR DE HUMO Y FUEGO, .....	18
1.6.3 ALARMA DE ENFRIAMIENTO, .....	18
1.6.4 ALARMA DE PUERTA ABIERTA,.....	19
1.6.5 ALARMA DE INTRUSO, .....	19
1.7 CONECCIÓN TÍPICA DE LAS ALARMAS EN EL ACU, .....	19
1.7.1 CONTACTO CERRADO,.....	19
1.7.2 CONTACTO ABIERTO, .....	19
1.8 FORMA DE OPERACIÓN DE LA UNIDAD DE CONTROL Y ALARMA (ACU),.....	21

1.8.1 STATUS DE UNA ALARMA, .....	21
1.8.2 TARJETA DE ALARMAS ( NT3P20FA ),.....	Pág 21
1.8.3 CONDICIÓN NORMAL, .....	21
1.8.4 CONDICIÓN DE ALARMA,.....	21

## CAPITULO II

### ESQUEMA GENERAL DEL SISTEMA

2.1 DESCRIPCIÓN DE LOS EQUIPOS, .....	22
2.1.1 EQUIPOS QUE SE UTILIZA DE LA COMPAÑÍA, .....	22
2.1.1.1 UNIDAD DE CONTROL Y ALARMA (ACU),.....	22
2.1.1.2 CONMUTADOR DE TELEFONÍA MÓVIL (MTX), .....	28
2.1.1.3 SERVIDOR ( ZETRON ),.....	29
2.1.1.4 BEEPER,.....	29
2.2 EQUIPOS A IMPLEMENTARSE, .....	29
2.2.1 TERMINAL,.....	29
2.2.2 MÓDEM,.....	30
2.3 DIAGRAMA DE BLOQUES,.....	30

## CAPITULO III

### PROCESAMIENTO GENERAL DEL SISTEMA

3.1 FUNCIONAMIENTO AUTOMATICO DEL SISTEMA, .....	32
3.2 MANTENIMIENTO DE LA BASE DE DATOS,.....	33

## CAPÍTULO IV

### IMPLEMENTACIÓN DEL PROYECTO

4.1 HARDWARE,.....	36
4.2 SOFTWARE, .....	37
4.2.1 MODULO MONITOREO DE ALARMAS, .....	37
4.2.2 MODULO ENVIO MENSAJE AL BEEPER, .....	40
4.2.3 MODULO DE ADMINISTRACION DE ALARMAS,.....	44

## **CAPITULO V**

### **PRUEBAS DEL SISTEMA**

5.1 PRUEBAS DEL SISTEMA, .....	Pág 46
5.2 OPERACIÓN DEL SISTEMA, .....	47
5.2.1 APERTURA DEL PUERTO, .....	48
5.2.2 DIRECCIONAMIENTO DEL PUERTO, .....	49
5.2.3 ENVIO DEL MENSAJE AL BEEPER, .....	50
5.2.4 TIEMPO DE VELOCIDAD DEL PROGRAMA, .....	51
5.3 EJEMPLO DE UNA PRUEBA EXITOSA, .....	52

### **CONCLUSIONES,**

### **ECOMENDACIONES,**

### **ANEXOS**

LISTADO DEL PROGRAMA, .....	59
MANUAL DEL USUARIO, .....	96

### **BIBLIOGRAFIA,**



**Biblioteca Central**

## CAPITULO I

### CARACTERÍSTICAS DE LAS ALARMAS DENTRO DE LA RED CELULAR

#### **1.1 CONFIGURACIÓN DE LA RED CELULAR EN GUAYAQUIL**

La red celular de la compañía Conecell - Guayaquil esta constituida por una estación base central y otras estaciones bases remotas, cada una de estas estaciones bases dan servicio a cierto sector creando lo que denominamos una celda, estas celdas se enlazan entre ellas vía microondas conectándose con la estación Central de Conmutación ó Switch. Fig. 1.1.

Cada una de estas bases poseen canales de control y canales de voz. Los canales de control se utilizan para establecer una llamada, y los canales de voz se utilizan para la conversación. Los canales de control entre dos celdas adyacentes son diferentes para evitar el solapamiento de frecuencia.

En la central se encuentran el conmutador telefónico móvil (MTX), que es el encargado de manejar las llamadas y datos que llegan de estaciones remotas hacia la central. Dentro de los datos que llegan a la central se encuentran las alarmas internas y externas, siendo las alarmas externas el interés de este proyecto, ya que son aquellas que están conectadas en estaciones bases remotas, donde no existe presencia directa de personal técnico y su activación es de enorme importancia para la seguridad del sistema.

Estas alarmas una vez que se han activado, deben ser reconocidas por la central para establecer el tipo de alarma y a que estación base pertenecen.

Toda la infraestructura que se requiere para monitoreo, detección y envío esta presente en esta tecnología celular, sin embargo para el caso de las alarmas externas de seguridad, no existe un aprovechamiento óptimo en su tratamiento, para este caso en particular.

Es así como este proyecto ha visto la necesidad de crear un sistema de software, capaz de obtener datos de un puerto de salida del MTX, tomando muy en cuenta su sincronización.

Luego comparar con una base de datos y filtrar las alarmas que en verdad nos interesen para luego ser enviadas vía módem hacia la estación base donde se encuentre la central del sistema busca personas (porta beeper), llamada zetron, y que luego enviará el mensaje de alarma activada hacia el o los supervisores indicados, para su posterior control.

A todo este camino que se debe recorrer es lo que hemos llamado configuración de la red celular del sistema, que se presenta en la figura 1.2.

La cuál nos muestra la disposición de las antenas remotas hacia la central de conmutación, además podemos observar los equipos que necesitamos algunos de los cuales deberán implementarse.

# CONFIGURACION DE LA RED CELULAR DEL SISTEMA

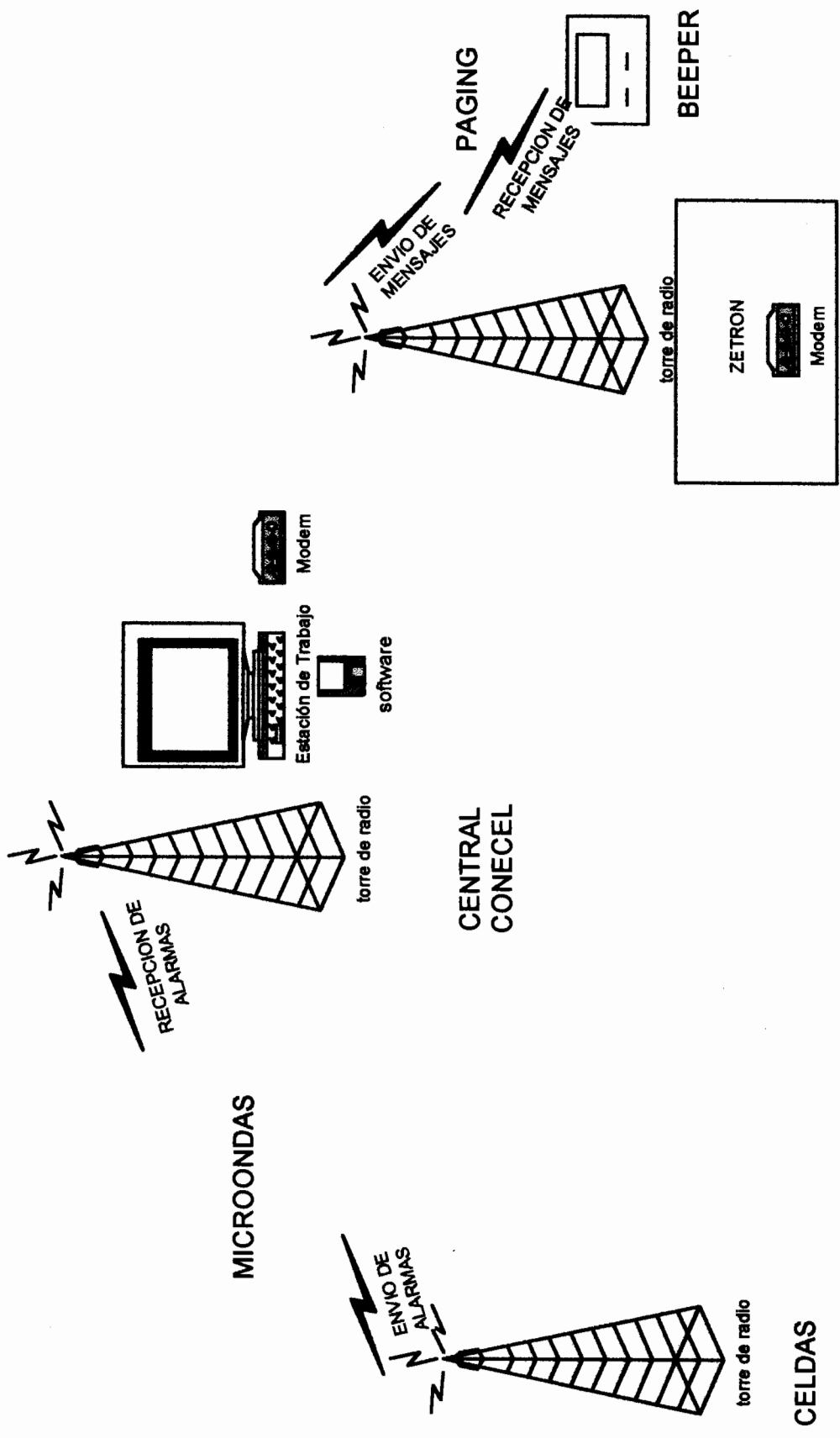


Fig. 1.2





## **1.2 CARACTERISTICAS DE LAS ALARMAS**

La característica fundamental de una alarma de seguridad esta en la forma de sensor, activarse, cualquier anomalía que se presente dentro de un sistema.

Las alarmas en su mayoría son de contacto abierto o cerrado accionado por relés y es la más utilizada; de haz de luz infrarroja que se activan cuando el haz es interrumpido; también existen las de tipo inalámbrico muy utilizados en entidades bancarias, y es activada por medio de una botonera de un circuito que emite una señal de onda corta.

### **1.2.1 CONCEPTO DE ALARMA**

Una alarma es una indicación explícita que indica, al operador, el estado anormal ó avería de un equipo que acarrea la degradación de la calidad, seguridad y confiabilidad de un servicio.

### **1.2.2 ALARMAS DE SEGURIDAD**

Son alarmas que nos permiten obtener un buen control y supervisión tanto para lo personal, bienes o equipos de una empresa que se tenga. Por lo tanto un buen sistema de alarmas de seguridad ayudará en la calidad y confiabilidad de cualquier servicio que se preste. Para una celda o central, las alarmas de seguridad son de tipo externo, es decir, son alarmas que se generan por causas externas al funcionamiento de la central o de la celda.

### **1.2.3 UNIDAD DE CONTROL Y ALARMA (ACU)**

En toda celda para poder garantizar su operación, tanto en su mantenimiento como en su seguridad, cuenta con una Unidad de Control y Alarma, ACU, que

se encuentra ubicado en el primer bastidor de la celda llamado equipo común, el ACU sensa continuamente todas las alarmas definidas; como son las alarmas internas que supervisan el buen funcionamiento y operación de la celda; y las alarmas externas que controlan la seguridad de la celda, tales como humo y fuego, puerta abierta, enfriamiento, energía.

### **1.3 CATEGORÍA DE LAS ALARMAS**

Por el grado de importancia que tienen las alarmas, se las ha clasificado en tres categorías:

#### **1.3.1 CATEGORÍA A1**

Son las de más alto grado de interés e importancia y estas son las alarmas internas.

#### **1.3.2 CATEGORÍA A2**

Es el segundo grado de importancia que puede tener una alarma, a esta categoría pertenecen las alarmas externas.

#### **1.3.3 CATEGORÍA A3**

Este es el último grado de interés que puede tener una alarma, a esta categoría pertenecen las alarmas de observación.

Cabe destacar que las alarmas a la que pertenecen a esta categoría son utilizadas en la central y no en las celdas, y son definidas más por los operadores de la central según el nivel de importancia y necesidad que se tenga. Es por esto que en algunas centrales, como por ejemplo la Central celular de Ericson hacen una subdivisión de esta categoría es decir, alarmas de categoría A3 y alarmas de observación.

## **1.4 CLASIFICACIÓN DE LAS ALARMAS**

Las alarmas se clasifican en: Alarmas internas, alarmas externas y alarmas de observación.

### **1.4.1 ALARMAS INTERNAS**

Son aquellas alarmas que nos indica la presencia de alguna anomalía que afecta el buen funcionamiento de un sistema.

### **1.4.2 ALARMAS EXTERNAS**

Son alarmas que se generan por causas externas a un sistema, a diferencia de las anteriores que se generan por causas internas a un sistema.

### **1.4.3 ALARMAS DE OBSERVACIÓN**

A diferencia de las alarmas externas e internas, este clase de alarma son definidas por el personal de operadores; y es utilizada para control de ambiente y suministros de las oficinas, como por ejemplo el sistema de aire acondicionado, falta papel en las computadoras, etc.

## **1.5 ATENCIÓN Y REPARACIÓN**

La atención y reparación que una alarma merece depende del tipo, importancia y categoría que esta tenga; así tenemos que las alarmas internas, que son las de primera categoría, son las de mayor importancia y su de atención y reparación debe ser en forma inmediata.

Las alarmas externas, de segunda categoría, su atención y reparación se la puede efectuar durante las 24 horas del día, y las alarmas de observación durante la semana.

## **1.6 DESCRIPCIÓN DE LAS ALARMAS EXTERNAS QUE SE PRESENTAN EN UNA CELDA**

A continuación describiremos las alarmas externas más importantes que son utilizadas:

- Energía
- Sensores de humo y fuego
- Enfriamiento
- Puerta abierta
- Intruso

### **1.6.1 ALARMA DE ENERGÍA**

Se generan cuando existe una falta de energía eléctrica pública. Cuando se presenta esta falta de energía, en centrales ó en edificios de gran interés y actividad, es reemplazada por generadores que proveen esta energía. Este tipo de alarma es de contacto abierto.

### **1.6.2 ALARMA SENSOR DE HUMO Y FUEGO**

Este es un solo tipo de alarma que cumple con dos funciones, sensar la presencia de humo y de fuego directo. Estas alarmas son de gran utilidad para evitar la pérdida de equipos o materiales por flagelo. Este tipo de alarma es utilizada en todas las celdas, debido a que en su mayoría las celdas son remotas.

### **1.6.3 ALARMA DE ENFRIAMIENTO**

Esta alarma se genera cuando sensa la variación de temperatura, de baja temperatura a alta temperatura. Cuando existe fallas en los equipos destinados a mantener el enfriamiento.

## **1.6.4 ALARMA DE PUERTA ABIERTA**

Ayuda en el control de acceso a áreas que son exclusivas solo para personal privado.

## **1.6.5 ALARMA DE INTRUSO**

Es una de las alarmas más utilizadas; su funcionamiento es por accionamiento de contactos, abierto o cerrado. Esta alarma se genera cuando existe el ingreso de persona(s), intruso(s), en zonas de seguridad y en horas no laborables. Dentro de esta característica se encuentra también las alarmas de asalto; son de tipo inalámbrico, es más utilizadas por entidades bancarias.

El accionamiento de esta alarma es manual y es efectuada por el personal que esta siendo objeto de un asalto.

## **1.7 CONECCIÓN TÍPICA DE LAS ALARMAS EN EL ACU**

Existe dos tipos de conexión típica de las alarmas; de contacto cerrado y de contacto abierto, que depende de las características del circuito con que esta fabricada una alarma, y de las necesidades o recursos que se tenga.

### **1.7.1 CONTACTO CERRADO**

Este tipo de conexión ambos extremos del circuito deben permanecer cerrados, de esta manera la alarma permanece desactivada (contacto cerrado). En el instante en que los extremos del circuito se abren la alarma se activa.

### 1.7.2 CONTACTO ABIERTO

En este tipo de conexión un extremo del lugar que se esta sensando, es puesto a tierra de esta manera el circuito permanece abierto y la alarma desactivada (contacto abierto), en el instante en que el extremo puesto a tierra es movlizado cierra el circuito y la alarma se activa.

La figura 1.1 presenta una conexión típica de alarmas, standard, en la tarjeta de entrada

### CONEXIÓN TÍPICA DE LAS ALARMAS

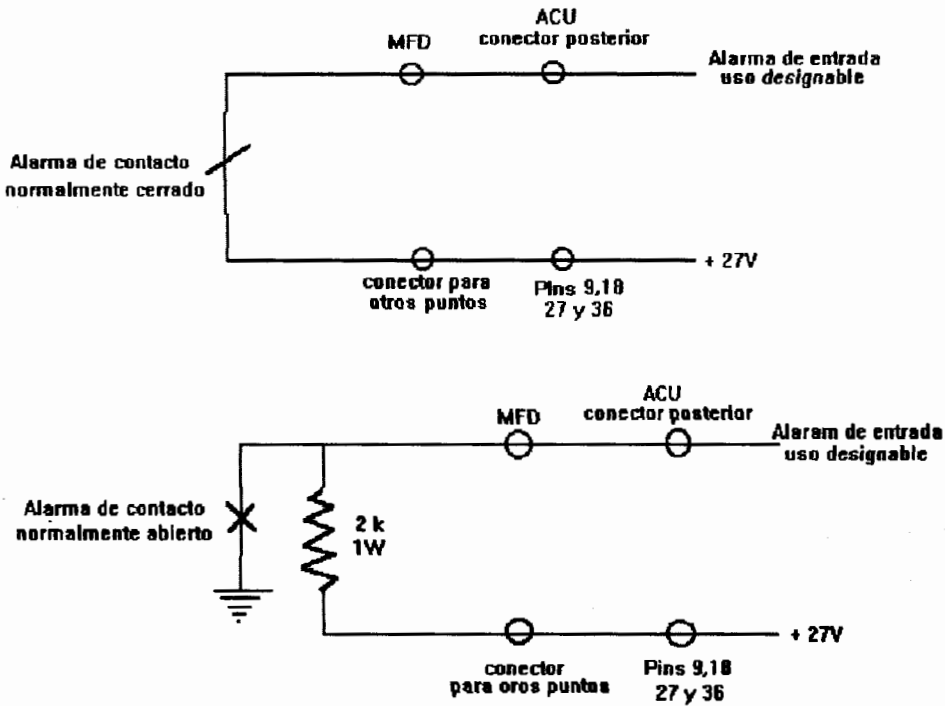


Fig. 1.1

## **1.8 FORMA DE OPERACIÓN DE LA UNIDAD DE CONTROL Y ALARMA (ACU)**

### **1.8.1 STATUS DE UNA ALARMA**

El status de una alarma indica el estado o condición presente de una alarma, el status lo da el ACU a través de la tarjeta NT3P20FA de alarmas que posee.

### **1.8.2 TARJETA DE ALARMAS ( NT3P20FA )**

El ACU, posee este tipo de tarjeta de entradas de alarmas. Una vez instalada el circuito de una alarma, es sensada en la central por medio del ACU a través de esta tarjeta. La tarjeta ( NT3P20FA ) tiene dos condiciones de funcionamiento; condición normal, o no alarma, y la de condición de alarma.

### **1.8.3 CONDICIÓN NORMAL**

La condición normal indica la no existencia de alarma, bajo esta condición, el circuito se mantiene cerrado con un nivel de voltaje alto (+ 27 V dc ). Es decir el circuito se activa con los + 27 V dc. cerrando así sus contactos y mantiene al circuito cerrado.

### **1.8.4 CONDICIÓN DE ALARMA**

La condición de alarma indica la existencia de la misma; una alarma existe cuando la condición de alarma es bajo (0 vol) ó circuito abierto (sin voltaje) en el punto de entrada de alarma.

La tarjeta de entrada de alarmas, reporta la presencia de una alarma únicamente cuando el nivel de voltaje se cambia.

## **CAPITULO II**

### **ESQUEMA GENERAL DEL SISTEMA**

#### **2.1 DESCRIPCIÓN DE LOS EQUIPOS**

A continuación describiremos los equipos que se utilizarán para el desarrollo de nuestro proyecto, empleando los recursos propios de la compañía operadora del servicio, que son utilizados por la compañía para brindar el servicio de telefonía celular; así como también aquellos equipos necesarios que deberán implementarse para complementar el desarrollo de nuestro proyecto.

##### **2.1.1 EQUIPOS QUE SE UTILIZA DE LA COMPAÑÍA**

Los equipos de la compañía que emplea el proyecto son los siguientes: el ACU, el conmutador de telefonía móvil (MTX), el servidor (zetron) utilizado para correo por Porta Beeper y un beeper.

###### **2.1.1.1 UNIDAD DE CONTROL Y ALARMA (ACU)**

Esta unidad se encuentra localizado en un bastidor llamado equipo común que se halla en la celda. El ACU es una Unidad de Control y Alarmas; es quien realiza un monitoreo discreto de las alarmas reportando y controlando el funcionamiento de la celda, el ACU concentra todos los puntos de entrada de alarma en la celda y pone al tanto a la central MTX de algún cambio de estado.

Cada celda puede tener un máximo de dos ACU, pero normalmente uno está instalado. La figura 2.1 muestra la ubicación de un ACU en el equipo común.



## UBICACIÓN DEL ACU EN EL EQUIPO COMÚN

### EQUIPO COMÚN

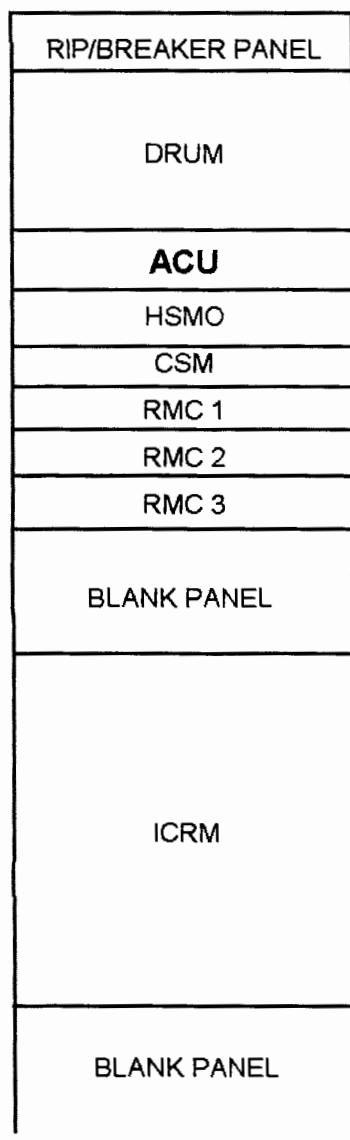


Fig. 2.1

Descripción de cada una de las partes que conforma el equipo común.

RIP/BREAKER PANEL	Panel de breakers
DRUM	Monitor de radio en modo dual
ACU	Unidad de control y alarma
HSMO	Oscilador master de alta estabilidad
CSM	Monitor de celda local
RMC	Receptores
BLANK PANEL	Panel vacío
ICRM	Módulo inteligente celular remoto

El ACU consta de los siguientes módulos:

- Una tarjeta microprocesador ( MPU ).
- Tarjeta de entrada y salida ( I/O ).
- Tarjetas de entradas de alarmas.
- Tarjetas de contacto de salida.
- Panel frontal.
- Panel posterior
- Módulo de fuente de alimentación.



**Biblioteca Central**

## TARJETA MICROPROCESADOR

El MPU almacena la configuración de las alarmas de entrada y de contacto de salida mediante el uso de memorias RAM; supervisa continuamente el estado de todos los alarmas, si una alarma es activada el MPU sensa el cambio de estado de la alarma y envía al MTX la condición actual de la misma.

## TARJETA DE ENTRADA Y SALIDA I/O

La tarjeta I/O se conecta a la tarjeta MPU mediante un cable tipo cinta de 44 conductores, a través de esta recibe señales de datos, dirección y control de bus. La tarjeta I/O suministra una circuitería decodificadora y conectores para las 16 tarjetas de I/O que van en los slots. También posee terminales para el voltaje de alimentación.

## TARJETAS DE ALARMAS DE ENTRADA Y TARJETAS DE CONTACTOS DE SALIDA

El ACU puede contener un máximo de 16 tarjetas de entrada de alarmas y de contactos de salida, cada tarjeta de entrada de alarma posee 16 puntos de entrada y cada tarjeta de contacto de salida controla 8 pares de contactos.

Por lo tanto el ACU puede controlar un total de 248 puntos de alarma. El mínimo requerimiento que se puede tener en el ACU es 6 tarjetas de entradas de alarmas y una tarjeta de contacto de salida.

La tabla 2.1 muestra la configuración de tarjetas que se puede tener en el ACU.

### CONFIGURACIÓN DE TARJETAS PARA EL ACU

Configuración	Tarjetas		Total de Alarma. De entrada		Contacto de salida (en pares)
	Alarma. Entrada	Contacto de salida	Alarma. Internas	Alarma. Externas	
Máxima	15	1	8	240	8
	14	2	8	224	16
	13	3	8	208	24
	12	4	8	192	32
Mínima	6	1	8	96	8

Fig. 2.1

En la figura 2.2 se presenta un esquema de la ubicación de los módulos que pertenecen al ACU.

### ESQUEMA DE LOS MÓDULOS PERTENECIENTES AL ACU.

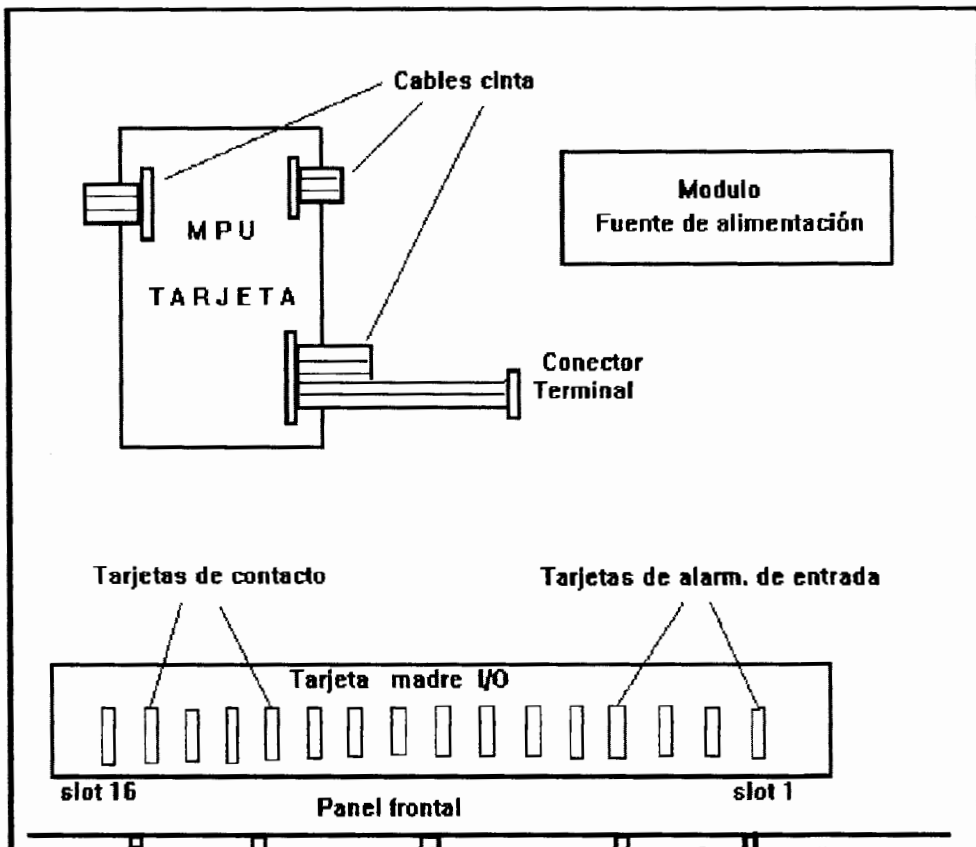


Fig. 2.2



## Biblioteca Central

### **CONFIGURACIÓN DEL JCP**

Una vez instaladas las alarmas, y para que puedan ser sensadas, deberán ser conectadas al ACU.

La tarjeta I/O de esta unidad tiene una capacidad de 16 slots para almacenar 16 tarjetas de alarmas; desde el slot 7 hasta el 12, se ubican las tarjetas que están destinadas solo para sensar las alarmas externas, estas se las asigna según su utilización.

En los slots 1 al 6 son usados para tarjetas de alarmas de entrada, alarmas internas, y son las que supervisan el status de las alarmas de la celda.

En los slots 13 y 15 pueden ir dos tarjetas para alarmas de entrada o dos tarjetas de contacto de salida.

En el slot 16, puede ser usada para mantener un tarjeta adicional de contacto de salida.

### **PANEL POSTERIOR**

Cada par de tarjetas se interconectan en un solo conector en la parte posterior del panel del ACU, por lo tanto tendremos un total de ocho conectores que tiene una denominación de J1, J2, J3,....., J8.

El mínimo requerimiento para cada celda, es un ACU que contenga una capacidad de seis tarjetas de alarmas de entrada con una tarjeta de alarma de contacto.

La tabla 2.2 presenta una distribución típica de las tarjetas de alarmas en la tarjeta I/O y sus respectivos conectores.

### **DISTRIBUCIÓN TÍPICA DE LAS TARJETAS DE ALARMAS**

Conectores	Tarjeta Madre I/O		Función
Panel posterior	Slot	Tipo de tarjeta I/O	
J1- Cable superior (azul rayado)	1	Tarjeta de alarma	Alarmas- RF bastidores 1 y 2
J1- Cable inferior	2	Tarjeta de alarma	Alarmas- RF bastidores 3 y 4
J2- Cable superior (azul rayado)	3	Tarjeta de alarma	Alarmas- RF bastidores 5 y 6
J2- Cable inferior	4	Tarjeta de alarma	Alarmas- Coax switch
J3- Cable superior (azul rayado)	5	Tarjeta de alarma	Alarmas- HSMO
J3- Cable inferior	6	Tarjeta de alarma	Alarmas- CSM y RMC
J4- Cable superior (azul rayado)	7	Tarjeta de alarma	Alarmas- de uso definido
J4- Cable inferior	8	Tarjeta de alarma	Alarmas- de uso definido
J5- Cable superior (azul rayado)	9	Tarjeta de alarma (opcional)	Alarmas- de uso definido
J5- Cable inferior	10	Tarjeta de alarma (opcional)	Alarmas- de uso definido
J6- Cable superior (azul rayado)	11	Tarjeta de alarma (opcional)	Alarmas- de uso definido
J6- Cable inferior	12	Tarjeta de alarma (opcional)	Alarmas- de uso definido
J7- Cable superior (azul rayado)	13	Alarma o contacto (opcional)	Alarmas/Control- de uso definido
J7- Cable inferior	14	Tarjeta de contacto	Control- Coax switch
J8- Cable superior (azul rayado)	15	Alarma o contacto (opcional)	Alarmas/Control- de uso definido
J8- Cable inferior	16	Tarjeta de contacto (opcional)	Controlador- de uso definido

Tabla 2.2

En la práctica se utiliza las tarjetas de los slots 7 y 8 , conector J4, con una capacidad para 32 diferentes tipos de alarmas.

#### **2.1.1.2 CONMUTADOR DE TELEFONIA MÓVIL (MTX)**

El MTX es una unidad de la central de conmutación, a este conmutador de telefonía móvil llegan todas las alarmas provenientes de los ACU de cada una de las celdas, y las alarmas que se encuentran instaladas en la misma central.

### **2.1.1.3 SERVIDOR ( ZETRON )**

Porta beeper es otro servicio que presta la compañía, el envío de un mensaje a un beeper lo realizan mediante el sistema de uso de operadoras, que envían los mensajes recibidos a través de un servidor ( zetron ).

Nuestro proyecto, accesa directamente al servidor para luego enviar el mensaje al beeper; de esta manera se ahorra tiempo y recursos de la compañía.

### **2.1.1.4 BEEPER**

Un beeper, o los beepers que la compañía considere necesario, se utilizará para que reciba el mensaje de las alarmas que están siendo sensadas por el supervisor.

El beeper debe tener el servicio de acceso directo, caso contrario no recibirá el mensaje respectivo.

## **2.2 EQUIPOS A IMPLEMENTARSE**

Los equipos necesarios para la implementación del proyecto son: un terminal y un módem.

### **2.2.1 TERMINAL**

Puede ser de tipo computadora o una laptop, en la que se instalará el software del proyecto, y efectuar el respectivo monitoreo de las alarmas.

El terminal, que se encuentra instalado a un puerto, serial, del MTX desde el cual se recibirán las alarmas que están siendo sensadas, deberá estar configurado a 2400 bps.

### **2.2.2 MÓDEM**

Es la interface entre el terminal y el beeper del supervisor, la configuración del módem para el enlace será de 1200 bps., 8 bits de datos, no paridad y 1 bit de parada ( 8,N,1 ).

Cabe destacar que el módem puede estar incorporado en el terminal.

### **2.3 DIAGRAMA DE BLOQUES**

Dentro de las instalaciones que conforma una celda, se tiene un equipo común en donde debe de haber una Unidad de Control y Alarma, ACU, que es el equipo que sensa continuamente todas las alarmas que están definidas, sean estas externas e internas.

Cada ACU transmite la información de las alarmas sensadas al MTX. El MTX forma parte de los equipos de conmutación de la Central celular, recibe la información de todas las alarmas de cada una de las celdas; que para el caso de la Central celular Guayaquil región costa se tiene hasta la presente 22 celdas.

Del MTX se toma todas las alarmas que son sensadas a través de uno de sus puertos, puertos seriales, por medio de un terminal (computadora) que procesará la información recibida para luego realizar el procedimiento de envío de las alarmas mediante el uso de un módem.

El módem es una interface entre el monitor y el servidor (zetrón ), para el envío del mensaje de las alarmas al beeper.

Cabe destacar que el número telefónico que se utiliza para el módem pertenece a la misma compañía que presta el servicio de telefonía celular.



Una vez que el servidor recibe el mensaje de las alarmas sensadas y además, el número del beeper que se le halla asignado al supervisor, comienza a realizar un procedimiento de búsqueda denominado ( PAGING ), por medio de este procedimiento ubica al beeper del supervisor quien a su vez recibirá el mensaje de las alarmas.

La figura 2.3 presenta un diagrama de bloques en la que se indica como está conformado el sistema.

### DIAGRAMA DE BLOQUES DEL SISTEMA

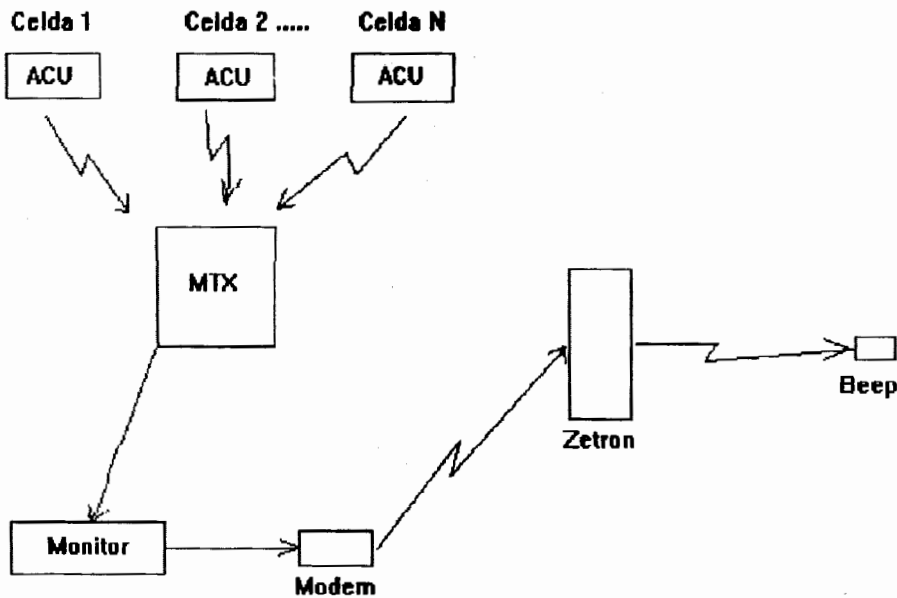


Fig. 2.3

## **CAPITULO III**

### **PROCESAMIENTO GENERAL DEL SISTEMA**

#### **3.1 FUNCIONAMIENTO AUTOMATICO DEL SISTEMA**

El sistema contará con un terminal que deberá estar conectado a un puerto, del MTX, desde el cual recibirá las alarmas de tipo de seguridad (alarmas externas); el puerto estará direccionado exclusivamente para recibir las alarmas externas.

Una vez encendido el sistema, puesta en funcionamiento, el software del programa abrirá el puerto al cual se encuentra conectado, de esta manera se podrá recibir todas las alarmas externas que se generen.

El software cuenta con una base de datos en la que se ha almacenado las alarmas de seguridad, codificadas por tipo de alarma y celda a la que pertenece; además cada una de ellas lleva un mensaje grabado para dar mayor facilidad de distinción del tipo de alarma que se presenta y facilitar la transmisión de la misma hacia el beeper del supervisor.

De esta manera; se podrá considerar que una alarma es de tipo de seguridad, una vez que se efectúe una comparación de las alarmas que se generan con las alarmas que se encuentran en la base de datos.

Desde el terminal se realizarán el monitoreo de las alarmas, emitirá un sonido indicando la presencia de una de ellas, después de cuatro segundos el sistema empezará a efectuar el procedimiento de envío de la alarma enlazándose con un terminal Dial Acces Pages (zetron ) para enviar la

alarma generada a un beeper del supervisor, este proceso es automático, y lo efectúa utilizando el módem como interface.

Si durante el procedimiento de envío de la alarma no se logra el enlace, debido a que se está utilizando las líneas telefónicas como medio de transmisión, el sistema recibirá el tono de ocupado ( no dialtone ), y volverá a realizar un nuevo intento de llamado. Este procedimiento lo realizará hasta que se halla efectuado el envío de la alarma al beeper.

Una vez que el proceso del envío de la alarma se halla efectuado, se recibirá como respuesta " Mensaje enviado ".

En caso de generarse mas de una alarma al mismo tiempo, el procedimiento de envío de la alarma hacia el beeper se realizará en forma serial.

En la pantalla del monitor se podrá observar procedimiento total por medio de tres ventanas. Es decir, en una ventana se observará las alarmas que llegan al sistema; en otra ventana denominada "cola de alarmas " se observará aquellas alarmas que serán enviadas al beeper del supervisor y en la tercera ventana se observará el procedimiento del envío de la alarma.

Además se cuenta con una ventana adicional para detener el proceso de envío de la alarma.

### **3.2 MANTENIMIENTO DE LA BASE DE DATOS**

A continuación presentaremos como poder realizar cambios a nuestra disposición de la base de datos, esto nos brinda una flexibilidad en el programa, para futuras ampliaciones de alarmas que la compañía desee incrementar.

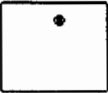



## BASE DE DATOS


CELDA	CODIGO	DESCRIPCION	CELNOM	+
19	97	ALARMA 24 PWR MI	24 PWR MI	
20	96	ALARMA 24 PWR MN	24 PWR MN	
20	97	ALARMA 24 PWR MI	24 PWR MI	
21	96	ALARMA 24 PWR MN	24 PWR MN	
21	97	ALARMA 24 PWR MI	24 PWR MI	

**MANTENIMIENTO DE LA BASE DE DATOS**

CELDA   
 CODIGO ALARMA   
 DESCRIPCION   
 CELNOM   
 CELDES



**Biblioteca Central**

En esta base de datos se encuentran registradas las alarmas externas de seguridad de cada una de las celdas. Esta pantalla permite darle mantenimiento a la base, es decir:

- Ingresar un nuevo registro
- Modificar un registro
- Eliminar un registro

## • INGRESO DE UN NUEVO REGISTRO

Pasos para ingresar un registro

1. Ingresar los datos en cada uno de los siguientes campos:

CELDA: Numero de la Celda

CODIGO ALARMA: Código de las alarmas externas

DESCRIPCION: Nombre o descripción de la alarma que se envía al Beeper

CELNOM: Nombre o descripción de la alarma que es capturada desde el MTX

CELDES: Nombre de la Celda.

2. En caso de una mala digitación presionar el botón LIMPIAR, si todo esta correcto pasar al siguiente paso

3. Presionar doble click sobre el botón TRANSMITIR.

4. Presionar doble click sobre el botón SALIR

## • MODIFICACION DE UN REGISTRO

Pasos para modificar un registro

1. Selección del Registro dando doble click sobre el mismo

2. Modificación de cualquiera de los campos:

CELDA

CODIGO ALARMA

DESCRIPCION

CELNOM

CELDES

3. Presionar doble click sobre el botón TRANSMITIR

4. Presionar doble click sobre el botón SALIR

- **ELIMINACION DE UN REGISTRO**

Pasos para eliminar un registro

1. Selección del Registro dando doble click sobre el mismo

2. Presionar doble click sobre el botón ELIMINAR.

3. Presionar doble click sobre el botón SALIR

## **CAPÍTULO IV**

### **IMPLEMENTACIÓN DEL PROYECTO**

#### **4.1 HARDWARE**

El hardware que el proyecto necesita para su ejecución es:

Procesador 386 o más.

4Mb. Mínimo RAM, 8MB. Recomendado.

Espacio en disco de 250 Kb.

Windows 3.11 o más.

Dos puertos seriales.

Modems de 2400 bps. mínimo (solo si la conexión es remota).

Existen dos tipos de conexiones; directa y remota.

**Conexión directa.** Para este tipo de conexión se utiliza un cable serial desde el PC al MTX.

**Conexión remota.** En cambio para este tipo de conexión se lo realiza a través de modems, desde el PC hacia el ZETRON.

Hay que aclarar en este momento, que no necesariamente las conexiones deben ser iguales a las del gráfico. En otras palabras se pueden tener.

- Dos conexiones remotas.
- Dos conexiones directas.
- Una directa hacia el ZETRON y otra remota hacia el MTX.
- O remota hacia el ZETRON y la otra directa hacia el MTX .

## 4.2 SOFTWARE

El software del proyecto se encuentra realizado en lenguaje VISUAL BASIC, en la que se observará en el apéndice A. Es un programa desarrollado para identificación de alarmas de seguridad mediante la utilización de tecnología celular.

Básicamente consiste de 3 módulos :

- Módulo de Monitoreo de las Alarmas.
- Módulo de Envío del mensaje al beeper.
- Módulo de Administración de Alarmas.

### 4.2.1 MODULO MONITOREO DE ALARMAS

Por medio de un puerto serial conectado al MTX, este módulo recibe las alarmas generadas en la celdas. Para la conexión con este puerto, el programa usa la lógica expuesta en fig 4.1:

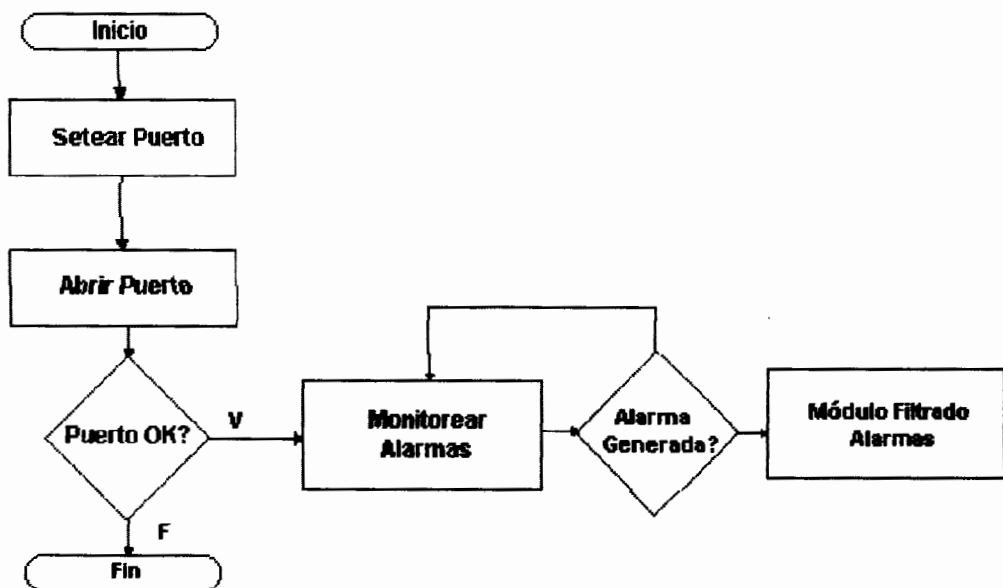


Fig. 4.1



Podemos ver en el diagrama todas las subrutinas utilizadas en este módulo.

## **Setear Puerto**

Obtiene el puerto a utilizar para la conexión con el MTX y los datos de configuración del mismo.

Los datos de configuración son tomados de un archivo de inicialización. Estos datos se encuentran en la sección [Puerto\_MTX] y son :

- **Port.**

Este parámetro indica que puerto va a utilizar :

1 - COM1

2 - COM2, etc.

- **Settings.**

Indica los parámetros de configuración del puerto especificado en el parámetro anterior. Estos parámetros son :

Velocidad

Paridad :

N - None

E - Even ó Par

O - Odd o Impar

- Número de bits.
- Bits de Parada.

Todos estos parámetros deben ir juntos, separados por una coma y en el orden especificado.

**Ejemplo :**

Settings = 2400,N,8,1

Donde,

Velocidad = 2400

*Paridad =NONE*

Número de Bits = 8

Bits de Parada = 1

- **Conección.**

Indica si la conexión al puerto es directa o remota.

D - Directa

R - Remota

Directa significa que está conectado localmente por medio de un cable serial.

Remota significa que se está usando un módem para la conexión. En este caso se debe especificar el número del módem en el parámetro siguiente.

- **Teléfono.**

Se especifica aquí el número del módem al cual marcar para hacer la conexión al puerto MTX. Este parámetro es requerido si el parámetro "Conección" ha sido seteado a Remota.

## **Abrir Puerto**

Esta subrutina abre el puerto especificado en el archivo de inicialización. Si el puerto es abierto exitosamente, se envía una señal de "Break" para habilitar el monitoreo de las alarmas.

Si hay algún problema en la apertura, se envía un mensaje de error y el programa termina.

## Monitorear Alarmas

Esta subrutina realiza un monitoreo constante del puerto, con el objetivo de verificar si alguna alarma se ha activado. Si alguna alarma se ha activado, ésta es pasada al módulo de Filtrado, para que éste decida si es necesaria enviarla.

### 4.2.2 MODULO ENVIO MENSAJE AL BEEPER

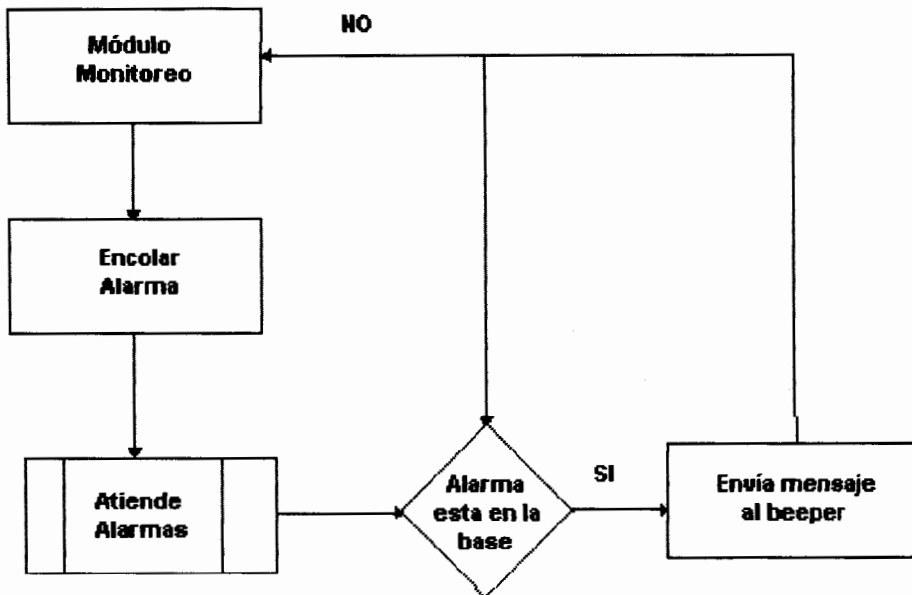


Fig. 4.2

En la figura 4.2, todas las alarmas son encoladas en una lista tipo FIFO (First In First Out).

Es decir, la primera alarma en llegar es la primera en ser atendida. El formato en que se graba es el siguiente :

Tipo - Celda - Nombre Alarma

Esta cola de alarmas aparece en la pantalla y se puede ver claramente todas las alarmas que se han activado y como el programa las va atendiendo.

## **Atiende Alarmas**

Este proceso continuamente esta atendiendo las alarmas que aparecen en la cola. Verifica que las alarmas esten en la base de datos, además de dar un tiempo prudente (que puede ser configurado) para detener el envío del mensaje.

El tiempo de espera, en segundos, se encuentra en el archivo de inicialización, bajo la sección [Alarma]. El parámetro es "Tiempo". Ejemplo :

[Alarma]

Tiempo=4

## **Envía mensaje al beeper**

Lee los seteos del puerto a utilizar para la comunicación con el Servicio PortaBeeper. Abre el puerto y si hay algun error vuelve a reintentar, hasta que el mensaje sea enviado. Se enlaza con el módem remoto. Envía los comandos necesarios para que el mensaje llegue al Supervisor.

El mensaje enviado al beeper se forma de :

1. Alarma
2. Nombre de alarma
3. Mensaje adicional. (Especificado en el archivo de inicialización).

Los comandos usados para el envío de mensajes son :

El programa remoto (ZETRON) después del enlace, responde con el mensaje "ID=".



**Biblioteca Central**

Se le envía una "M".

ZETRON responde con el mensaje "Logon granted.", y después con el mensaje "Page = ".

Se le envía el código del beeper (especificado en el archivo de inicialización).

ZETRON responde con el Nombre del usuario del beeper.

Se le envía el mensaje de la alarma activada.

ZETRON responde con "Transaction acknowledge".

ZETRON responde con otro mensaje "Page = ". Para que pueda seguir enviando otro mensaje al mismo o a otro beeper.

Se le envía el comando "bye".

ZETRON responde con "Transaction acknowledge".

Los seteos del puerto se encuentran en la sección [Puerto\_Beeper] y son :

- **Port.** Este parámetro indica que puerto va a utilizar :
  - 1 - COM1
  - 2 - COM2, etc.
- **Settings.** Indica los parámetros de configuración del puerto especificado en el parámetro anterior. Estos parámetros son :

Velocidad

Paridad :

N - None

E - Even ó Par

Odd o Impar

- Número de bits.
- Bits de Parada.

Todos estos parámetros **deben ir juntos**, separados por una coma y en el orden especificado.

**Ejemplo :**

Settings = 1200,E,8,1

**Donde,**

Velocidad = 1200

Paridad =EVEN

Número de Bits = 8

Bits de Parada = 1

- **Conección.** Indica si la conexión al puerto es directa o remota.

D - Directa

R - Remota

**Directa** significa que esta conectado localmente por medio de un cable serial.

**Remota** significa que se esta usando un módem para la conexión. En este caso se debe especificar el número del módem en el parámetro siguiente.

**Teléfono.** Se especifica aquí el número del módem al cual marcar para hacer la conexión al ZETRON. Este parámetro es requerido si el parámetro "Conexión" ha sido seteado a Remota.

- **Código.** Especifica el código del beeper, el cual recibirá las alarmas.

### 4.2.3 MODULO DE ADMINISTRACION DE ALARMAS

En este módulo se especifican cuales alarmas serán enviadas al beeper y cuales serán ignoradas. Junto a la alarma se especifica un nombre descriptivo, el cual será enviado como parte del mensaje. El programa accesa a una base de datos cuyas tablas se presentan a continuación :

#### ALARMAS

Campo	Descripción
CELDA	Código de la celda
CODIGO	Código de la alarma
CELNOM	Nombre de la alarma
CELDESC	Identificación de la celda

Este módulo permite hacer Consulta, Mantenimiento, Eliminación e Ingreso de Alarmas. Todas las alarmas que son ingresadas por este módulo, son consideradas para el envío de mensaje al beeper. Por el contrario, las alarmas que no aparecen serán ignoradas y no serán enviadas.

## **CAPITULO V**

### **PRUEBAS Y OPERACIÓN DEL SISTEMA**

#### **5.1 PRUEBAS DEL SISTEMA**

Para poder realizar adecuadamente el software de este sistema hubo que realizar pruebas en la central de conmutación, ya que necesitábamos entablar una conversación con el MTX, puesto que la compañía posee instalados terminales que dan información, pero que no es posible modificarla.

Nuestro mayor problema estuvo entonces en la abertura del puerto, del que solo conocíamos que dando un CONTROL BREAK nos respondía y podíamos acceder a la información.

Lo más lógico sería entonces dentro del programa dar ese control break para abrir el puerto serial del MTX, lo cuál no sucedió, entonces pensamos en otra posibilidad conectando un terminal de conmutación en serie con nuestra computadora que mediante un software identifique ese comando de control y poder llevarlo al programa lo cuál lo hicimos, pero no respondió.

Decidimos entonces chequear en una computadora, utilizando un simulador como es el HiperTerminal, que significado tenía el control break y encontramos qué se trataba de un comando con una característica especial de duración de 150 mseg aproximadamente, con lo cuál utilizado dentro del programa pudimos finalmente abrir el puerto.



Una vez que es abierto el puerto serial observamos una gama de información a la cuál podemos acceder, ya que nuestro interés se centra en las alarmas externas entonces hay que hacer una correcta elección de la dirección que debemos tomar.

Existe un comando LOGUTIL, que nos presenta el estado actual de todas las alarmas del tipo externas que se hayan activado, con un formato tal que podemos conocer de que celda se trata y de que tipo de alarma es, por lo tanto nuestro programa puede ser capaz de detectar estos campos y compararlos con la base de datos, como efectivamente ocurrió.

Luego que es capturada la alarma y filtrada en la base de datos, necesitábamos enviarla al beeper, para lo cuál era necesario entablar otro enlace, pero esta vez del tipo "acceso remoto directo", es decir desde la Central de Conmutación por medio de la computadora con un software adecuado, accesar remotamente a la Central de beepers (Zetron), sin necesidad de pasar por las operadoras.

Para el envío del mensaje al beeper, la sincronización entre nuestro programa y el software del sistema beeper fue un factor muy importante que hubo que superar, ya que en principio nuestro programa era muy rápido y no teníamos acceso al sistema. Por lo tanto dentro del programa se creó una subrutina considerando éste desface.

## **5.2 OPERACIÓN DEL SISTEMA**

El software está diseñado en lenguaje VISUAL BASIC, cuyo menú presenta una variedad de comandos que facilitan al usuario el diseño de un programa.

Un programa diseñado en VISUAL BASIC presenta una imagen más compacta, facilitando su lectura y entendimiento, que a diferencia de otros lenguajes dan como resultado un programa demasiado extenso.

Las pruebas efectuadas a nuestro proyecto podemos definirlas en cuatro etapas:

### 5.2.1 APERTURA DEL PUERTO

Generalmente, los operadores de conmutación, lo realizan desde un terminal mediante el siguiente procedimiento:

Operador presiona : CONTROL + BREAK

Respuesta : ?

Operador digita : hx

Respuesta : Conecell Guayaquil-Ecuador MTX04BW

Si el operador desea recibir todas las alarmas que se están sensando:

Operado digita : mtxalm all

o si desea solo las alarmas externas :

Operador digita : mtxext.

Este procedimiento lo realizamos desde nuestro programa; con la finalidad de evitar errores en la digitación y acelerar el proceso.

CONTROL + BREAK es un procedimiento que solamente se lo realiza desde los terminales de conmutación, según el tipo de terminal, otro terminal independiente del sistema no puede efectuarlo.

Este inconveniente lo consideramos dentro de nuestro programa, CONTROL + BREAK lo realizamos mediante el comando CONTROL + BREAK = TRUE , con el VISUAL BASIC, dentro de una subrutina considerándose un tiempo de 150 mseg.

Además la digitación de hx y mttext forman parte de la subrutina.

### **5.2.2 DIRECCIONAMIENTO DEL PUERTO**

Del MTX se puede obtener todas las alarmas que están siendo sensadas, tanto externas como internas, el direccionamiento del puerto es con el objetivo de obtener solamente las alarmas externas por lo tanto el puerto debe estar bien direccionado, si el puerto no esta bien direccionado se podrá observar en la pantalla en la ventana de "puerto" lo siguiente :

||||||||||||||||||||||||||||||||||||||||

o también

wwwwwwwwwwwwww

Para evitar un mal direccionamiento, por los operadores, lo realizamos desde nuestro programa de la siguiente manera :

logutil :

deldevice Map1

addrp Map1 MTX 140

stardev Map1

Con MTX 140 obtenemos las alarmas externas, y con stardev Map1 aparecerán en la pantalla las alarmas externas por Map1.

Para hacer una prueba de esta parte del programa; se procede a bajar un ACU, este procedimiento se lo efectúa por medio de software, y que consiste en activar dicho ACU. Para las pruebas efectuadas se tomo como ejemplo el ACU de la celda centro.

### **5.2.3 ENVÍO DEL MENSAJE AL BEEPER**

Se lo realiza utilizando como interface un módem configurándolo a 1200 bps, 8 bits de datos, paridad par y 1 bit de parada. Esta configuración se la utilizó debido a que la compañía operadora ofrece el servicio de Porta Beeper, y opera con los datos anteriormente mencionados por medio de un terminal de paging ( ZETRON ).

Para el envío del mensaje al beeper, de las alarmas que son sensadas, se tiene una base de datos; en esta base se tiene grabado un mensaje de distinción para cada una de las alarmas externas de seguridad.

Esta base de datos sirve para realizar un filtrado de todas las alarmas que se sensan con respecto a las que deben ser informadas al supervisor por medio del beeper.

En la pantalla se tiene una ventana denominada "colas de las alarmas" , con el fin de poder observar las alarmas que van a ser enviadas. Una vez de haber sido enviada esta desaparece.

Para poder controlar el proceso de enlace hacia el beeper, se tiene en la pantalla otra ventana denominada " beeper " ; y se realiza de la siguiente manera :

Programa : atdt 09402566

Respuesta : CONET 1200

Programa : # del beeper

Respuesta : login. # del beeper. Nombre del usuario

Programa : M

Programa : se envía el mensaje de alarma

Respuesta : se observa el mensaje que es enviado

Respuesta : mensaje enviado

Cabe destacar que para el servicio de Porta Beeper se tiene un tiempo de diez segundos para enviar un mensaje, es por esto que en la base de datos se tiene grabado los mensajes a enviarse.

Este procedimiento tiene una duración aproximadamente de quince segundos a partir desde el enlace con el ZETRON, y lo realizamos dentro de una subrutina considerando un tiempo de 10000 mseg.

#### **5.2.4 TIEMPO DE VELOCIDAD DEL PROGRAMA**

Cabe destacar que el tiempo respuesta con respecto a la velocidad del programa, es un punto muy importante.

En muchas ocasiones no se podía obtener los resultados deseados debido a que el programa avanza más rápido sin dar paso al tiempo de respuesta.

Este problema, crítico, se lo resolvió mediante retardos de tiempo dentro de las subrutinas de cada proceso.

### **5.3 EJEMPLO DE UNA PRUEBA EXITOSA**

Se realizaron más de una prueba para asegurarnos que nuestro programa funcione adecuadamente.

Tomando como ejemplo la celda CENTRO , citaremos una de las pruebas realizadas. Cabe destacar que esta celda anteriormente fue objeto de robo de la sirena que indicaba la presencia de intruso, por lo tanto mantenía activada dicha alarma, motivo por el cual se la tomo para efecto de prueba de nuestro proyecto. Una vez puesto en operación nuestro proyecto, se procedió a sensar la alarma por medio del MTX, a través del ACU de dicha celda, cabe recordar que el procedimiento es automático. Como podemos observar en la siguiente figura 5.1

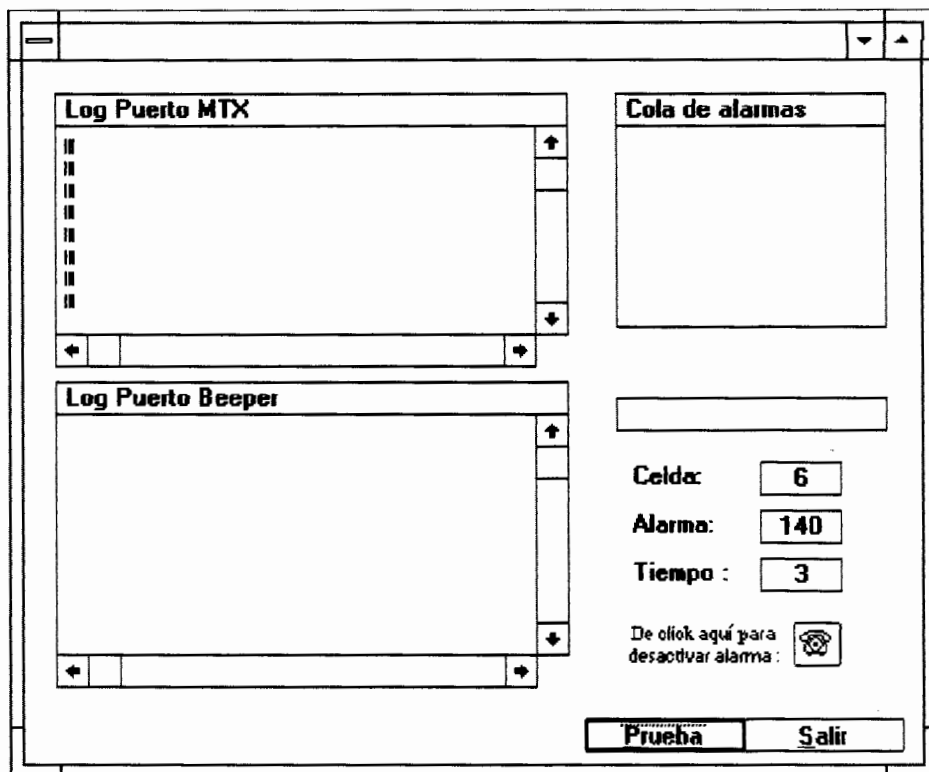


Fig. 5.1

El monitoreo de la alarma se la observa en el recuadro "Log puerto MTX", así como también en el recuadro "Alarma generada" podemos observar que en la Celda 6 se ha generado la alarma 140, este código pertenece a la alarma SIRENA CENTRO y podemos observarla en el recuadro "Cola de alarmas", este recuadro se lo utiliza para que se vayan acumulando todas las alarmas que se van generando. Luego procede a realizarse el envío del mensaje a un beeper de destino, al del supervisor, para lo cual primero se enlaza con la central de beeper's, el envío del mensaje lo realiza en forma automática y directa, es decir no se pasa por las operadoras de la central de beeper's, marcándose el número respectivo de la central 9402566, como podemos observar en la figura 5.2.

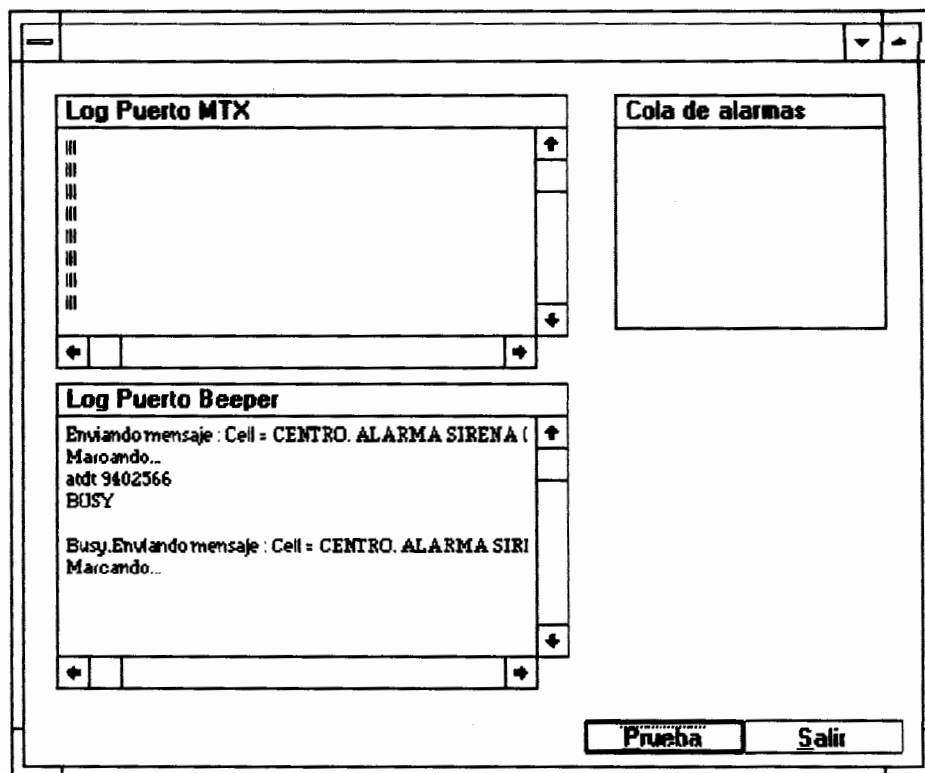


Fig. 5.2

Una vez que consigue enlazarse con la central, se procede a enviarse el mensaje al beeper, para el ejemplo se utilizó el de la Ing. Priscila Mite de Conecell, como podemos observar en la figura 5.3.



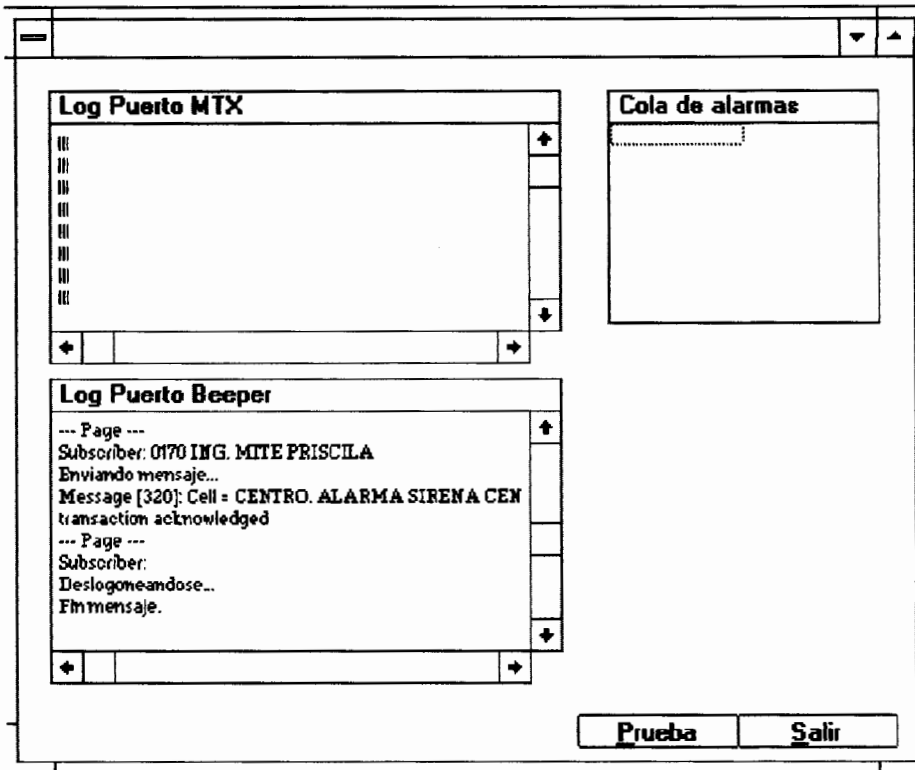


Fig. 5.3

Cabe recalcar que todo este procedimiento es completamente automático, además si en un primer intento de enviar de mensaje no se lo logra, el procedimiento se repite hasta que se lo halla realizado.

Para asegurarnos de que la alarma sensada no vuelva a repetirse en su respectivo envío, en el recuadro de colas de alarmas, esta desaparece.

Si existe más de una alarma generada, en dicho recuadro se van acumulando en el orden que se vayan presentando, y en ese mismo orden son enviadas, una alarma no desaparece hasta que no se halla realizado su respectivo envío.

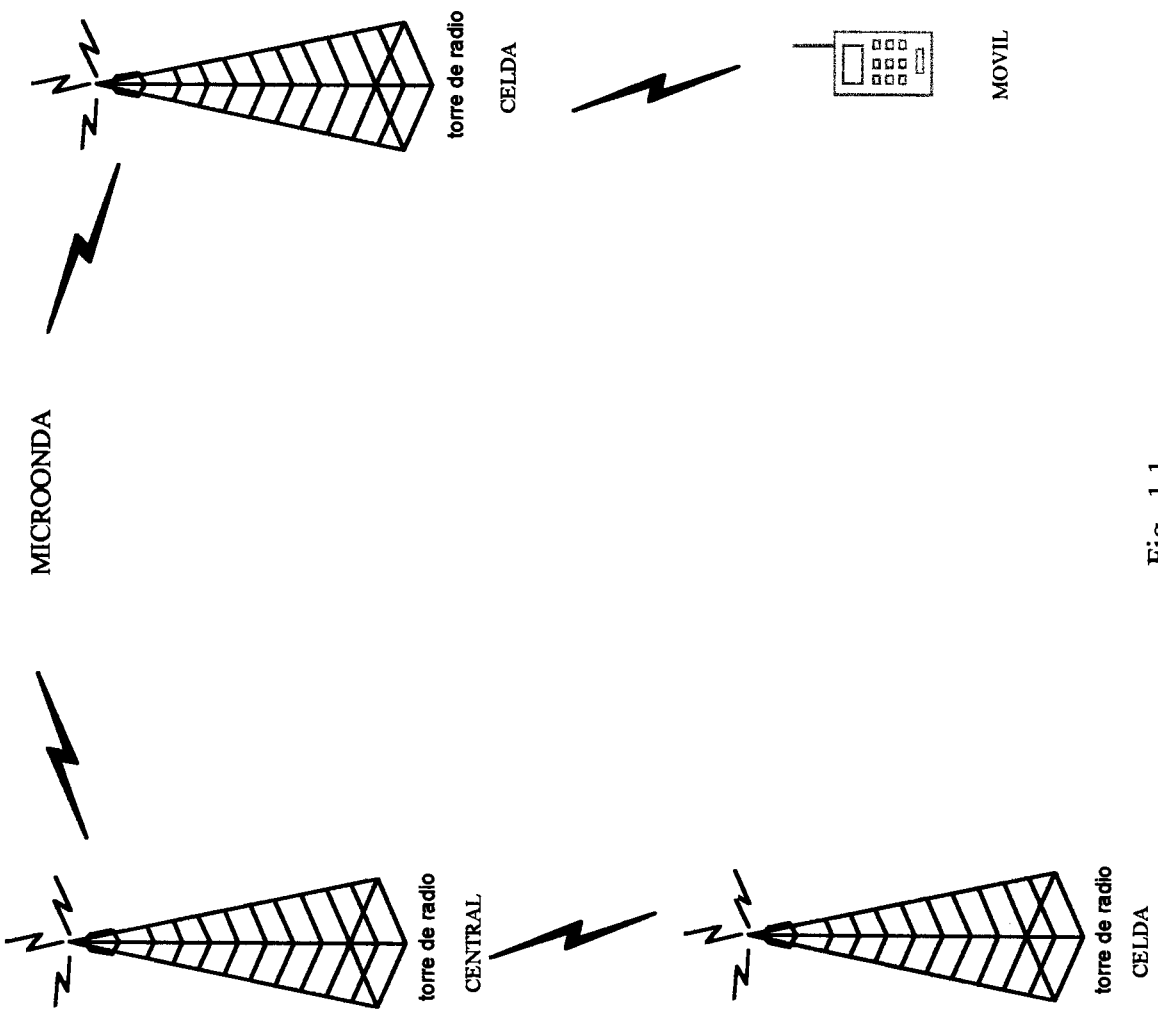


Fig. 1.1

Otra forma de realizar pruebas, consiste en hacer que el ACU de cualquier celda sea desactivada, conocido por los operadores de la central de conmutación como “bajar el acu”, pero esto no podía realizarse en forma continua para efecto de prueba, ya que esto afectaría el buen desenvolvimiento de control y operación en la central.



**Biblioteca Central**

## **CONCLUSIONES**

Con el desarrollo de este proyecto, podemos concluir que:

- Todo proyecto debe tener una aportación al desarrollo técnico, sea este de interés institucional, privado o comunitario. Nuestro proyecto logra con ésta aportación y no solo a un ente privado; si no que también deja conocimientos de la tecnología celular, y una aplicación que se puede tener de ella, muy provechoso para futuros profesionales en telecomunicaciones.
- Las alarmas, en general, son las que nos presenta el estado actual de cualquier sistema puesta en servicio tanto local como remoto; pues de ellas depende la calidad y confiabilidad de los sistemas que intervienen. Por lo tanto, la calidad y tipos de alarmas que se utilicen deben ser de gran exactitud y de respuesta inmediata.
- La telefonía celular es uno de los sistemas mas avanzados que actualmente se tiene, su stock de alarmas esta conformado por una Unidad Central de Alarmas ( ACU ); que supervisa el funcionamiento total de todos los equipos que intervienen. Tiene capacidad de 248 puntos de alarmas, de estos 248 puntos de alarmas, 240 son para las alarmas externas, las demás son para las alarmas internas, lo que demuestra una seguridad total de funcionamiento.
- El monitoreo, control e identificación de una alarma así como la información de la misma a un supervisor de seguridad empleando la tecnología celular, cuyo medio de transmisión son las señales de microondas, es la técnica más segura y difícil de alterar, superando así a los sistemas convencionales de seguridad ya conocidos.

- Podríamos también concluir; que la gran capacidad de alarmas que posee el sistema celular, puede ser también utilizado para control y supervisión de seguridad en los lugares donde se hallen instaladas las celdas, como por ejemplo en los edificios, donde estén instaladas las celdas, en el centro de la ciudad, de esta manera se generaría nuevos ingresos económicos para la compañía.
- La aplicación de este proyecto puede extenderse a otras entidades privadas que deseen tener un sistema de seguridad confiable, o también a la otra compañía operadora del servicio celular, OTECELL. Para Otecell, habría que realizarse un estudio de el tipo de hardware y software que poseen para hacer los cambios y aplicaciones respectivos.

## **RECOMENDACIONES**

- Se debe tener en cuenta que el puerto al cual este conectado el terminal; este bien direccionado, de lo contrario no se podrá obtener las alarmas que el terminal necesita.
- Tener presente que la velocidad de enlace entre el terminal y el mtx es de 2400 bps; y la interface entre el terminal y el zetrón, para llegar al beeper, es un módem a 1200 bps.
- La base de datos debe ser actualizada acorde al crecimiento de las celdas y sus respectivas alarmas.
- El beeper a utilizar debe tener el servicio de acceso directo, caso contrario no podrá recibir los mensajes respectivos.

***ANEXOS***

---

## PROGRAMA

### ABRIR ARCHIVO DE BASE DE DATOS

```
Sub abrirArchivoDeDatos ()
    Dim I As Integer
    dtaTopico.Connect = ""
    dtaTopico.DataBaseName = DataBaseName
    dtaTopico.ReadOnly = False
    dtaTopico.Exclusive = False
    dtaTopico.RecordSource = "select * from alarmas order by celda"
    dtaTopico.Refresh
    For I = 0 To dtaTopico.Database.TableDefs.Count - 1
        cboAlarma.AddItem dtaTopico.Database.TableDefs(I).Name
    Next
    cboAlarma.Text = "(ninguna)"
End Sub
```

### CARGA GRID

```
Sub cargargrd ()
    abrirArchivoDeDatos
    cboAlarma.Text = "ALARMAS"
    RellenarGrid cboAlarma.Text
End Sub
```

### CBO\_ALARMAS\_CLICK

```
Sub cboAlarmas_Click ()
    cboAlarma.Text = "ALARMAS"
```



RellenarGrid cboAlarma.Text

End Sub

RELLENAR GRID

Sub RellenarGrid (ByVal NombreDeLaTabla As String)

Dim I As Integer, AnchoDeCelda As Integer

'Inicializar el conjunto de registros

dtaTopico.RecordSource = NombreDeLaTabla

'Establecer las Cabeceras de las columnas

grdDatos.Cols = dtaTopico.Database(NombreDeLaTabla).Fields.Count

grdDatos.Row = 0

For I = 0 To dtaTopico.Database(NombreDeLaTabla).Fields.Count - 1

grdDatos.Col = I

grdDatos.Text = dtaTopico.Database(NombreDeLaTabla).Fields(I).Name

grdDatos.ColWidth(I) = TextWidth(grdDatos.Text) + 100

Next I

dtaTopico.Refresh

'Determinar el numero de registros y establecer el tamaño del Grid

dtaTopico.Recordset.MoveFirst

'grdDatos.Row = 0

Do While Not dtaTopico.Recordset.EOF

grdDatos.Rows = grdDatos.Rows + 1

grdDatos.Row = grdDatos.Rows - 1

For I = 0 To dtaTopico.Database(NombreDeLaTabla).Fields.Count - 1

grdDatos.Col = I

'Almacenar un blanco si el dato del campo es nulo

If IsNull(dtaTopico.Recordset(I).Value) Then

grdDatos.Text = ""

Else



```

    grdDatos.Text = dtaTopico.Recordset(l).Value
End If
'Asegurarse de que todos los datos se ven
AnchoDeCelda = TextWidth(grdDatos.Text) + 100
If AnchoDeCelda > grdDatos.ColWidth(l) Then
    grdDatos.ColWidth(l) = AnchoDeCelda
End If
Next l
    dtaTopico.Recordset.MoveNext
Loop
End Sub

```

#### CDM\_ABRIR

```

Sub cmdabrir_click ()
    cboAlarma.Clear
    dlogo.DefaultExt = "MDB"
    dlogo.FileName = "Topico.mdb"
    dlogo.Filter = "Bases de datos (*.MDB|*.MDB|Todos os archivos (*.*)|*.*)"
    dlogo.Flags = OFN_NOREADONLYRETURN
    dlogo.Action = 1 'Abrir el cuadro de dialogo
    If dlogo.FileName = "" Then Exit Sub
    DataBaseName = dlogo.FileName
    abrirArchivoDeDatos
End Sub

```

#### CDM\_BUSCAR

```

Sub cmdbuscar_Click ()
    'Dim sql As String
    'Dim qry As dynaset

```

```

'Dim db_beeper As Database
'AbrirBase
'Set db_beeper = OpenDatabase("C:\VB\TOPICO.MDB", True, False)
'sql = "Select descripcion from alarmas "
'sql = sql + " where codigo = " + Text2.text + " and celda = " + Text1.text +
",."
'Set qry = db_beeper.CreateDynaset(sql)
'While Not qry.EOF
    Text3.text = qry("descripcion")
    qry.MoveNext
'Wend
'qry.Close
'db_beeper.Close
text1.Text = ""
text2.Text = ""
text4.Text = ""
text5.Text = ""
text6.Text = ""
End Sub

```

## CDMLIMPIAR

```

Sub cmdlimpiar_Click ()
    Dim sql As String
    Dim qry As dynaset
'AbrirBase
sql = "delete * from alarmas "
sql = sql + " where codigo = " + text2.Text + " and celda = " + text1.Text +
",."
db_beeper.Execute (sql)

```

```
cargargrd
text1.Text = ""
text2.Text = ""
text4.Text = ""
text5.Text = ""
text6.Text = ""
End Sub
```

## CDMTRANS

```
Sub cmdTrans_Click ()
    Dim t As table
    If text1.Text = "" Then
        MsgBox "El campo CELDA es obligatorio"
        text1.SetFocus
        Exit Sub
    End If
    If text2.Text = "" Then
        MsgBox "El campo CODIGO ALARMA es obligatorio"
        text2.SetFocus
        Exit Sub
    End If
    If text4.Text = "" Then
        MsgBox "El campo DESCRIPCION es obligatorio"
        text4.SetFocus
        Exit Sub
    End If
    If text5.Text = "" Then
        MsgBox "El campo CELNOM es obligatorio"
        text5.SetFocus
    End If
End Sub
```

```
Exit Sub
End If
If text6.Text = "" Then
    MsgBox "El campo CELDESC es obligatorio"
    text6.SetFocus
Exit Sub
End If

Set t = db_beeper.OpenTable("alarmas")
t.AddNew
t("celda") = text1.Text
t("codigo") = text2.Text
t("descripcion") = text4.Text
t("celnom") = text5.Text
t("celdesc") = text6.Text
t.Update
t.Close
cargargrd
text1.Text = ""
text2.Text = ""
text4.Text = ""
text5.Text = ""
text6.Text = ""
End Sub
```

FORM

```
Sub Form_Load ()
    Me.Width = 7200
    Me.Height = 5280
```

```

Me.Top = 0
Me.Left = 0
cboAlarma.Clear
dlogo.DefaultExt = "MDB"
dlogo.FileName = DataBaseName
dlogo.Filter = "Bases de datos (*.MDB|*.MDB|Todos los archivos (*.*)|*.*)"
dlogo.Flags = OFN_FILEMUSTEXIST
'dlogo.Action = 1 'Abrir el cuadro de dialogo
If dlogo.FileName = "" Then Exit Sub
abrirArchivoDeDatos Using ("Puerto_")
cboAlarma.Text = "ALARMAS"
RellenarGrid cboAlarma.Text el puerto
End Sub

```

grdDATOS

```

Sub grdDatos_DblClick ()
grdDatos.Col = 0
text1.Text = grdDatos.Text
grdDatos.Col = 1
text2.Text = grdDatos.Text
grdDatos.Col = 2
text4.Text = grdDatos.Text
grdDatos.Col = 3
text5.Text = grdDatos.Text
grdDatos.Col = 4
text6.Text = grdDatos.Text
End Sub

```

ABRIR PUERTO



**Biblioteca Central**

**Function AbrirPuertos () As Integer**

**Dim ret As Integer**

**Dim field As String**

**On Error Resume Next**

**Me.Show**

**Me.MousePointer = 11**

**AbrirPuertos = False**

**'Abriendo puerto MTX**

**field\$ = Space(129)**

**ret = GetPrivateProfileString("Puerto\_MTX", "Port", "", field\$, 128, IniFile\$)**

**If ret <= 0 Then**

**MsgBox "Debe especificar el puerto asignado para MTX"**

**Me.MousePointer = 0**

**Exit Function**

**End If**

**comm\_beeper.CommPort = CInt(Trim(Left(field\$, ret)))**

**comm\_beeper.InputLen = 0**

**'Seteando parametros coneccion**

**field\$ = Space(129)**

**ret = GetPrivateProfileString("Puerto\_MTX", "Settings", "", field\$, 128,**

**IniFile\$)**

**If ret <= 0 Then**

**MsgBox "Debe especificar los seteos para abrir puerto MTX"**

**Me.MousePointer = 0**

**Exit Function**

**End If**

**comm\_puerto.InputLen = 0**

**comm\_puerto.Settings = Trim(Left(field\$, ret))**

**comm\_puerto.RTSEnable = True**

```

comm_puerto.DTREnable = True
comm_puerto.Handshaking = MSCOMM_HANDSHAKE_XONXOFF
comm_puerto.PortOpen = False
Err = 0
comm_puerto.PortOpen = True
If Err <> 0 Then
    MsgBox "Error al abrir Puerto MTX - COM" +
Format$(comm_puerto.CommPort, "0"), 16, "Abriendo Puerto MTX"
    Me.MousePointer = 0
    Exit Function
End If

'Abriendo puerto Beeper
field$ = Space(129)
ret = GetPrivateProfileString("Puerto_Beeper", "Port", "", field$, 128,
IniFile$)
If ret <= 0 Then
    MsgBox "Debe especificar el puerto asignado para Beeper"
    Me.MousePointer = 0
    Exit Function
End If
comm_beeper.CommPort = CInt(Trim(Left(field$, ret)))
'Seteando parametros coneccion
field$ = Space(129)
ret = GetPrivateProfileString("Puerto_Beeper", "Settings", "", field$, 128,
IniFile$)
If ret <= 0 Then
    MsgBox "Debe especificar los seteos para abrir puerto Beeper"
    Me.MousePointer = 0

```



Exit Function

End If

comm\_beeper.Settings = Trim(Left(field\$, ret))

comm\_beeper.RTSEnable = True

comm\_beeper.DTREnable = True

comm\_beeper.Handshaking = MSCOMM\_HANDSHAKE\_XONXOFF

comm\_beeper.PortOpen = False

Err = 0

comm\_beeper.PortOpen = True

If Err <> 0 Then

MsgBox "Error al abrir Puerto MTX = COM" +

Format\$(comm\_beeper.CommPort, "0"), 16, "Abriendo Puerto Beeper"

Me.MousePointer = 0

Exit Function

End If

AbrirPuertos = True

Me.MousePointer = 0

End Function

ATENDER MENSAJES

Sub AtenderMensajes ()

Dim mensaje As String

Dim alarma As String

Dim celda As String

Dim ret As Integer

Dim field As String

Dim celnom As String

If Cerrar Then Exit Sub

If lst\_alarmas.ListCount > 0 Then

If Not busy Then

busy = True

lst\_alarmas.ListIndex = 0

temp = lst\_alarmas.Text

k = InStr(1, temp, "-")

If k > 0 Then

alarma = Trim(Mid(temp, 1, k - 1)).

temp = Trim(Mid(temp, k + 1, Len(temp) - k))

k = InStr(1, temp, "-")

If k > 0 Then

celda = Trim(Mid(temp, 1, k - 1))

celnom = Trim(Mid(temp, k + 1, Len(temp) - k))

fx\_get\_mensaje alarma, celda, celnom, mensaje

If Trim(mensaje) <> "" Then

field\$ = Space(129)

ret = GetPrivateProfileString("Alarma", "Tiempo", "", field\$, 128, IniFile\$)

If ret < 0 Then

field\$ = "10"

Else

field\$ = Trim(Left(field\$, ret))

End If

AlarmTime = CInt(field\$)

fx\_set\_contador

Cancelar = False

SetearAlarma alarma, celda

Do

DoEvents

Loop While AlarmTime > 0 And Cancelar = False

CancelarAlarma

```
    If AlarmTime = 0 Then
        If EnviarMsgBeeper(mensaje, fx_get_codigo_beeper()) Then
            Ist_alarmas.RemoveItem 0
        End If
    End If
Else
    'No pararle bola a la alarma q' no esta en la base
    Ist_alarmas.RemoveItem 0
End If
End If
End If
busy = False
End If
End If
End Sub
```

#### CANCELAR ALARMA

```
Sub CancelarAlarma ()
    tmr_msg.Enabled = False
    Cancelar = True
    lbl_alarma.Caption = ""
    lbl_contador.Caption = ""
    lbl_celda.Caption = ""
    pnl_alarma.Visible = False
End Sub
```

#### CERRAR PUERTO

```
Sub CerrarPuertos ()
    On Error Resume Next
```

```

Dim resp$, linea$
Me.MousePointer = 11
comm_puerto.Output = "qa" + Chr$(13)
Do
    DoEvents
Loop While comm_puerto.OutBufferCount > 0
Do
    resp = comm_puerto.Input
    DoEvents
Loop While comm_puerto.InBufferCount > 0
comm_puerto.PortOpen = False
comm_beeper.PortOpen = False
Err = 0
Me.MousePointer = 0
End Sub

```

#### ENVIAR MENSAJE AL BEEPER

```

Function EnviarMsgBeeper (mensaje As String, beeper As String) As Integer
    Dim resp$, flag%, pos%, old$, c$, field$, ret%
    Dim Enviado%
    Me.MousePointer = 11
    Enviado = False
    If beeper = "" Then
        MsgBox "Especifique beeper", 16, "Mensaje"
        GoTo Otros
    End If
    txt_beeper.Text = txt_beeper.Text + "Enviando mensaje : " + mensaje + "." +
Chr$(13) + Chr$(10)
    comm_beeper.InputLen = 1

```

```

field$ = Space(129)
ret = GetPrivateProfileString("Puerto_Beeper", "Coneccion", "", field$, 128,
IniFile$)
If ret < 0 Then
    field$ = "R"
Else
    field$ = Trim(Left(field$, ret))
End If
If UCase$(field$) = "R" Then
    'Marca el numero
    field$ = Space(129)
    ret = GetPrivateProfileString("Puerto_Beeper", "Telefono", "", field$, 128,
IniFile$)
    If ret < 0 Then
        GoTo NoTelf
    Else
        field$ = Trim(Left(field$, ret))
    End If
    txt_beeper.Text = txt_beeper.Text + "Marcando..." + Chr$(13) + Chr$(10)
    envio = "atdt " + field$
    comm_beeper.Output = envio + Chr$(13)
    Do
        DoEvents
    Loop While comm_beeper.OutBufferCount > 0
    'Recibe el mismo numero
    old = "0"
    resp = ""
    i = 0
    Do

```

```

DoEvents
old = c
r = comm_beeper.Input
If r <> "" Then
    c = r
    i = i + 1
    If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
End If
Loop Until old = Chr$(13) And c = Chr$(10)
If Cerrar Then GoTo Fin
txt_beeper.Text = txt_beeper.Text + resp + Chr$(13) + Chr$(10)
pos = InStr(1, resp, "NO CARRIER")
If pos > 0 Then GoTo NoCarrier
pos = InStr(1, resp, "NO DIAL TONE")
If pos > 0 Then GoTo NoDialtone
pos = InStr(1, resp, "BUSY")
If pos > 0 Then GoTo busy
pos = InStr(1, resp, envio)
If pos <= 0 Then GoTo Otros
'Recibe el connect
old = "0"
resp = ""
i = 0
Do
    DoEvents
    old = c
    r = comm_beeper.Input
    If r <> "" Then
        c = r

```

```

    i = i + 1
    If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
End If
Loop Until old = Chr$(13) And c = Chr$(10)
If Cerrar Then GoTo Fin
txt_beeper.Text = txt_beeper.Text + resp + Chr$(13) + Chr$(10)
    pos = InStr(1, resp, "NO CARRIER")
If pos > 0 Then GoTo NoCarrier
pos = InStr(1, resp, "NO DIALTONE")
If pos > 0 Then GoTo NoDialtone
pos = InStr(1, resp, "BUSY")
If pos > 0 Then GoTo busy
pos = InStr(1, resp, "CONNECT")
If pos <= 0 Then GoTo Otros
    'Recibe el ID
old = "0"
resp = ""
i = 0
Do
    DoEvents
    old = c
    r = comm_beeper.Input
    If r <> "" Then
        c = r
        i = i + 1
        If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
    End If
Loop Until (old = Chr$(13) And c = Chr$(10)) Or c = "="
txt_beeper.Text = txt_beeper.Text + resp + Chr$(13) + Chr$(10)

```

```

    pos = InStr(1, resp, "NO CARRIER")
If pos > 0 Then GoTo NoCarrier
    pos = InStr(1, resp, "NO DIAL TONE")
If pos > 0 Then GoTo NoDialTone
    pos = InStr(1, resp, "BUSY")
If pos > 0 Then GoTo busy
    pos = InStr(1, resp, "IDLE")
If pos <= 0 Then GoTo Otros
End If

'Envia la M
txt_beeper.Text = txt_beeper.Text + "Logoneandose..." + Chr$(13) +
Chr$(10)
comm_beeper.Output = "M" + Chr$(13)
Do
    DoEvents
Loop While comm_beeper.OutBufferCount > 0
'Recibe el logon
old = "0"
resp = ""
i = 0
Do
    DoEvents
    old = c
    r = comm_beeper.Input
If r <> "" Then
    c = r
    i = i + 1
    If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c

```



```

End If
Loop Until old = Chr$(13) And c = Chr$(10)
    txt_beeper.Text = txt_beeper.Text + resp + Chr$(13) + Chr$(10)
    pos = InStr(1, resp, "NO CARRIER")
If pos > 0 Then GoTo NoCarrier
pos = InStr(1, resp, "NO DIAL TONE")
If pos > 0 Then GoTo NoDialtone
pos = InStr(1, resp, "BUSY")
If pos > 0 Then GoTo busy
pos = InStr(1, resp, "logon")
If pos <= 0 Then GoTo Otros
    'Recibe el page
old = "0"
resp = ""
i = 0
Do
    DoEvents
    old = c
    r = comm_beeper.Input
    If r <> "" Then
        c = r
        i = i + 1
        If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
    End If
Loop Until old = Chr$(13) And c = Chr$(10)
    txt_beeper.Text = txt_beeper.Text + resp + Chr$(13) + Chr$(10)
    pos = InStr(1, resp, "NO CARRIER")
If pos > 0 Then GoTo NoCarrier
pos = InStr(1, resp, "NO DIAL TONE")

```

```

If pos > 0 Then GoTo NoDialtone
pos = InStr(1, resp, "BUSY")
If pos > 0 Then GoTo busy
pos = InStr(1, resp, "Page")
If pos <= 0 Then GoTo Otros
    comm_beeper.Output = beeper + Chr$(13)
Do
    Sleep(129)
    DoEvents
Loop While comm_beeper.OutBufferCount > 0
'Recibe el suscriber
old = "0"
resp = ""
i = 0
Do
    DoEvents
    old = c
    r = comm_beeper.Input
    If r <> "" Then
        c = r
        i = i + 1
        If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
    End If
Loop Until old = Chr$(13) And c = Chr$(10)
txt_beeper.Text = txt_beeper.Text + resp + Chr$(13) + Chr$(10)
pos = InStr(1, resp, "NO CARRIER")
If pos > 0 Then GoTo NoCarrier
pos = InStr(1, resp, "NO DIALTONE")
If pos > 0 Then GoTo NoDialtone
pos = InStr(1, resp, "BUSY")

```

```

If pos > 0 Then GoTo busy
pos = InStr(1, resp, "Subscriber")
If pos <= 0 Then GoTo Otros
    'Enviar el mensaje
    txt_beeper.Text = txt_beeper.Text + "Enviando mensaje..." + Chr$(13) +
Chr$(10)
    field$ = Space(129)
    ret = GetPrivateProfileString("Alarma", "Mensaje_Adicional", "", field$, 128,
IniFile$)
    If ret < 0 Then
        field$ = "Urgente."
    Else
        field$ = Trim(Left(field$, ret))
    End If
    comm_beeper.Output = mensaje + "." + field$ + Chr$(13)
Do
    DoEvents
Loop While comm_beeper.OutBufferCount > 0
    'Recibir el Message
old = "0"
resp = ""
i = 0
Do
    DoEvents
    old = c
    r = comm_beeper.Input
    If r <> "" Then
        c = r
        i = i + 1
    End If
Loop While i <= 10

```

```
If (old = Chr$(13) And c = Chr$(10)) And (i >= 0 And i <= 2) Then c =
```

```
""
```

```
If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
```

```
End If
```

```
Loop Until old = Chr$(13) And c = Chr$(10)
```

```
txt_beeper.Text = txt_beeper.Text + resp + Chr$(13) + Chr$(10)
```

```
pos = InStr(1, resp, "NO CARRIER")
```

```
If pos > 0 Then GoTo NoCarrier
```

```
pos = InStr(1, resp, "NO DIALTONE")
```

```
If pos > 0 Then GoTo NoDialtone
```

```
pos = InStr(1, resp, "BUSY")
```

```
If pos > 0 Then GoTo busy
```

```
pos = InStr(1, resp, "Message")
```

```
If pos <= 0 Then GoTo Otros
```

```
'Recibir el acknowledged
```

```
old = "0"
```

```
resp = ""
```

```
i = 0
```

```
Do
```

```
DoEvents
```

```
old = c
```

```
r = comm_beeper.Input
```

```
If r <> "" Then
```

```
    c = r
```

```
    i = i + 1
```

```
    If (old = Chr$(13) And c = Chr$(10)) And (i = 0 Or i = 1) Then c = ""
```

```
    If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
```

```
End If
```

```
Loop Until old = Chr$(13) And c = Chr$(10)
```

```

txt_beeper.Text = txt_beeper.Text + resp + Chr$(13) + Chr$(10)
pos = InStr(1, resp, "NO CARRIER")
If pos > 0 Then GoTo NoCarrier
pos = InStr(1, resp, "NO DIALTONE")
If pos > 0 Then GoTo NoDialtone
pos = InStr(1, resp, "BUSY")
If pos > 0 Then GoTo busy
pos = InStr(1, resp, "acknowledged")
If pos <= 0 Then GoTo Otros
    Enviado = True
'Recibir otro page
old = "0"
resp = ""
i = 0
Do
    DoEvents
    old = c
    r = comm_beeper.Input
    If r <> "" Then
        c = r
        i = i + 1
        If (old = Chr$(13) And c = Chr$(10)) And (i = 0 Or i = 1) Then c = ""
        If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
    End If
Loop Until old = Chr$(13) And c = Chr$(10)
    txt_beeper.Text = txt_beeper.Text + resp + Chr$(13) + Chr$(10)
pos = InStr(1, resp, "NO CARRIER")
If pos > 0 Then GoTo NoCarrier
pos = InStr(1, resp, "NO DIALTONE")

```



**Biblioteca Central**

If pos > 0 Then GoTo NoDialtone

pos = InStr(1, resp, "BUSY")

If pos > 0 Then GoTo busy

pos = InStr(1, resp, "Page")

If pos <= 0 Then GoTo Otros

'Recibir otro Subscriber

old = "0"

resp = ""

i = 0

Do

DoEvents

old = c

r = comm\_beeper.Input

If r <> "" Then

c = r

i = i + 1

If (old = Chr\$(13) And c = Chr\$(10)) And (i = 0 Or i = 1) Then c = ""

If c >= Chr\$(32) And c <= Chr\$(128) Then resp = resp + c

End If

Loop Until old = Chr\$(13) And c = Chr\$(10)

txt\_beeper.Text = txt\_beeper.Text + resp + Chr\$(13) + Chr\$(10)

pos = InStr(1, resp, "NO CARRIER")

If pos > 0 Then GoTo NoCarrier

pos = InStr(1, resp, "NO DIALTONE")

If pos > 0 Then GoTo NoDialtone

pos = InStr(1, resp, "BUSY")

If pos > 0 Then GoTo busy

pos = InStr(1, resp, "Subscriber")

If pos <= 0 Then GoTo Otros



Biblioteca Central

```

'Deslogonearse
txt_beeper.Text = txt_beeper.Text + "Deslogoneandose..." + Chr$(13) +
Chr$(10)
comm_beeper.Output = "bye" + Chr$(13)
Do
    DoEvents
Loop While comm_beeper.OutBufferCount > 0
'Recibir el acknowledged
old = "0"
resp = ""
i = 0
Do
    DoEvents
    old = c
    r = comm_beeper.Input
    If r <> "" Then
        c = r
        i = i + 1
        If (old = Chr$(13) And c = Chr$(10)) And (i = 0 Or i = 1) Then c = ""
        If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
    End If
Loop Until old = Chr$(13) And c = Chr$(10)
pos = InStr(1, resp, "NO CARRIER")
If pos > 0 Then
    txt_beeper.Text = txt_beeper.Text + "Fin mensaje." + Chr$(13) +
Chr$(10)
    GoTo Bien
End If
pos = InStr(1, resp, "NO DIALTONE")

```

If pos > 0 Then GoTo NoDialtone

pos = InStr(1, resp, "BUSY")

If pos > 0 Then GoTo busy

GoTo Otros

Bien:

Me.MousePointer = 0

txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Mensaje  
enviado OK."

EnviarMsgBeeper = True

Exit Function

NoCarrier:

Me.MousePointer = 0

If Enviado Then

txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Mensaje  
enviado OK."

EnviarMsgBeeper = True

Else

txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "No Carrier."

EnviarMsgBeeper = False

End If

Exit Function

busy:

Me.MousePointer = 0

If Enviado Then

txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Mensaje enviado  
OK."

EnviarMsgBeeper = True

Else

txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Busy."



**EnviarMsgBeeper = False**

**End If**

**Exit Function**

**NoDialtone:**

**Me.MousePointer = 0**

**If Enviado Then**

**txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Mensaje enviado  
OK."**

**EnviarMsgBeeper = True**

**Else**

**txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "No Dialtone."**

**EnviarMsgBeeper = False**

**End If**

**Exit Function**

**Invalid:**

**Me.MousePointer = 0**

**If Enviado Then**

**txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Mensaje enviado  
OK."**

**EnviarMsgBeeper = True**

**Else**

**txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Codigo de Beeper  
invalido."**

**EnviarMsgBeeper = False**

**End If**

**Exit Function**

Otros:

**Me.MousePointer = 0**

**If Enviado Then**

**txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Mensaje  
enviado OK."**

**EnviarMsgBeeper = True**

**Else**

**txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Error al  
enviar mensaje."**

**EnviarMsgBeeper = False**

**End If**

**Exit Function**

NoTelf:

**Me.MousePointer = 0**

**If Enviado Then**

**txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Mensaje  
enviado OK."**

**EnviarMsgBeeper = True**

**Else**

**txt\_beeper.Text = txt\_beeper.Text + Chr\$(13) + Chr\$(10) + "Telefono no  
ha sido especificado."**

**EnviarMsgBeeper = False**

**End If**

**Exit Function**

Fin:

**Me.MousePointer = 0**

**EnviarMsgBeeper = True**

**End Function**

## FX\_ADD\_ALARMA

```
Sub fx_add_alarma (alarma As String, celda As String, mensaje As String)
    Dim texto As String
    texto = Trim(alarma) + "-" + Trim(celda) + "-" + Trim(mensaje)
    lst_alarmas.AddItem texto
End Sub
```

## OBTENER EL CODIGO DEL BEEPER

```
Function fx_get_codigo_beeper () As String
    Dim codigo$, ret%
    'Leyendo informacion de usuario
    codigo$ = Space(129)
    ret = GetPrivateProfileString("Puerto_Beeper", "Codigo", "", codigo$, 128,
IniFile$)
    If ret < 0 Then
        codigo$ = ""
    Else
        codigo$ = Trim(Left(codigo$, ret))
    End If
    fx_get_codigo_beeper = codigo$
End Function
```

## INICIAR PUERTO MTX

```
Function IniciarPuertoMTX () As Integer
    Dim resp$, linea As String, i As Integer
    Dim usuario$, field$, ret%
    IniciarPuertoMTX = False
    Me.MousePointer = 11
    field$ = Space(129)
```

```

ret = GetPrivateProfileString("Puerto_MTX", "Coneccion", "", field$, 128,
IniFile$)
If ret < 0 Then
    field$ = "D"
Else
    field$ = Trim(Left(field$, ret))
End If
If Asc(field$) = Chr$(32) Then
    If UCase$(field$) = "R" Then
        'Marca el número
        field$ = Space(128)
        ret = GetPrivateProfileString("Puerto_MTX", "Telefono", "", field$, 128,
IniFile$)
        If ret < 0 Then GoTo NoTelMTX
        GoTo NoTelMTX
    Else
        field$ = Trim(Left(field$, ret))
    End If
    If field$ = "D" Then GoTo busq
    txt_status.Text = txt_status.Text + "Marcando..." + Chr$(13) +
Chr$(10)
    envio = "atdt " + field$
    comm_puerto.Output = envio + Chr$(13)
    Do
        DoEvents
    Loop While comm_puerto.OutBufferCount > 0
    'Recibe el mismo numero
    old = "0"
    resp = ""
    i = 0
    Do

```

```

DoEvents
old = c
r = comm_puerto.Input
If r <> "" Then
    c = r
    i = i + 1
    If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
End If
Loop Until old = Chr$(13) And c = Chr$(10)
If Cerrar Then GoTo FinMTX
txt_status.Text = txt_status.Text + resp + Chr$(13) + Chr$(10)
pos = InStr(1, resp, "NO CARRIER")
If pos > 0 Then GoTo NoCarrierMTX
pos = InStr(1, resp, "NO DIALTONE")
If pos > 0 Then GoTo NoDialtoneMTX
pos = InStr(1, resp, "BUSY")
If pos > 0 Then GoTo busyMTX
pos = InStr(1, resp, envio)
If pos <= 0 Then GoTo OtrosMTX
'Recibe el connect
old = "0"
resp = ""
i = 0
Do
    DoEvents
    old = c
    r = comm_puerto.Input
    If r <> "" Then
        c = r

```

```

        i = i + 1
        If c >= Chr$(32) And c <= Chr$(128) Then resp = resp + c
    End If
Loop Until old = Chr$(13) And c = Chr$(10)
    If Cerrar Then GoTo FinMTX
        txt_status.Text = txt_status.Text + resp + Chr$(13) + Chr$(10)
        pos = InStr(1, resp, "NO CARRIER")
        If pos > 0 Then GoTo NoCarrierMTX
        pos = InStr(1, resp, "NO DIALTONE")
        If pos > 0 Then GoTo NoDialtoneMTX
        pos = InStr(1, resp, "BUSY")
        If pos > 0 Then GoTo busyMTX
        pos = InStr(1, resp, "CONNECT")
        If pos <= 0 Then GoTo OtrosMTX
    End If
'Break 1
comm_puerto.Break = True
For i = 1 To 10000
Next
comm_puerto.Break = False
'Recibiendo ?
Do
    DoEvents
Loop While comm_puerto.OutBufferCount > 0
Do
    resp = comm_puerto.Input
    Do
        linea = QuitarLinea(resp)
        txt_status.Text = txt_status.Text + linea + Chr$(10)
    
```

```

    Loop While resp <> ""
Loop While comm_puerto.InBufferCount > 0
    'Break 2
comm_puerto.Break = True
For i = 1 To 10000
Next
comm_puerto.Break = False
'Recibiendo ?
Do
    DoEvents
Loop While comm_puerto.OutBufferCount > 0
Do
    resp = comm_puerto.Input
    Do
        linea = QuitarLinea(resp)
        txt_status.Text = txt_status.Text + linea + Chr$(10)
    Loop While resp <> ""
Loop While comm_puerto.InBufferCount > 0
    'Leyendo informacion de usuario
usuario$ = Space(129)
ret = GetPrivateProfileString("General", "Usuario", "", usuario$, 128,
IniFile$)
If ret < 0 Then
    usuario$ = ""
Else
    usuario$ = Trim(Left(usuario$, ret))
End If
    'Enviando hx
comm_puerto.Output = "hx" + Chr$(13)

```

```

Do
    DoEvents
Loop While comm_puerto.OutBufferCount > 0
Do
    resp = comm_puerto.Input
    Do
        linea = QuitarLinea(resp)
        txt_status.Text = txt_status.Text + linea + Chr$(10)
    Loop While resp <> ""
Loop While comm_puerto.InBufferCount > 0
    'LogUtil
comm_puerto.Output = "logutil" + Chr$(13)
Do
    DoEvents
Loop While comm_puerto.OutBufferCount > 0
Do
    resp = comm_puerto.Input
    Do
        linea = QuitarLinea(resp)
        txt_status.Text = txt_status.Text + linea + Chr$(10)
    Loop While resp <> ""
Loop While comm_puerto.InBufferCount > 0
    'StopDev
comm_puerto.Output = "stopdev " + usuario$ + Chr$(13)
Do
    DoEvents
Loop While comm_puerto.OutBufferCount > 0
Do
    resp = comm_puerto.Input

```



```
Do
    linea = QuitarLinea(resp)
    txt_status.Text = txt_status.Text + linea + Chr$(10)
Loop While resp <> ""
Loop While comm_puerto.InBufferCount > 0
'DelDevice
comm_puerto.Output = "deldevice " + usuario$ + Chr$(13)
Do
    DoEvents
Loop While comm_puerto.OutBufferCount > 0
Do
    resp = comm_puerto.Input
    Do
        linea = QuitarLinea(resp)
        txt_status.Text = txt_status.Text + linea + Chr$(10)
    Loop While resp <> ""
Loop While comm_puerto.InBufferCount > 0
'Addrep
comm_puerto.Output = "addrep " + usuario$ + " MTX 140" + Chr$(13)
Do
    DoEvents
Loop While comm_puerto.OutBufferCount > 0
Do
    resp = comm_puerto.Input
    Do
        linea = QuitarLinea(resp)
        txt_status.Text = txt_status.Text + linea + Chr$(10)
    Loop While resp <> ""
Loop While comm_puerto.InBufferCount > 0
```

```

'Startdev
comm_puerto.Output = "startdev " + usuario$ + Chr$(13)
Do
  DoEvents
Loop While comm_puerto.OutBufferCount > 0
Do
  Me.Pointer = 0
  resp = comm_puerto.Input
  Do
    linea = QuitarLinea(resp)
    txt_status.Text = txt_status.Text + linea + Chr$(10)
  Loop While resp <> ""
Loop While comm_puerto.InBufferCount > 0
IniciarPuertoMTX = True
Me.MousePointer = 0
Exit Function
NoCarrierMTX:
Me.MousePointer = 0
txt_status.Text = txt_status.Text + Chr$(13) + Chr$(10) + "No carrier."
Exit Function
busyMTX: leyendoDatos
Me.MousePointer = 0
txt_status.Text = txt_status.Text + Chr$(13) + Chr$(10) + "Busy."
Exit Function
NoDialtoneMTX:
Me.MousePointer = 0
txt_status.Text = txt_status.Text + Chr$(13) + Chr$(10) + "No Dialtone."
Exit Function
InvalidMTX:
Me.MousePointer = 0

```

```
txt_status.Text = txt_status.Text + Chr$(13) + Chr$(10) + "Comando  
invalido."
```

```
Exit Function
```

```
OtrosMTX:
```

```
Me.MousePointer = 0
```

```
txt_status.Text = txt_status.Text + Chr$(13) + Chr$(10) + "Error  
encontrado."
```

```
Exit Function
```

```
NoTelMTX:
```

```
Me.MousePointer = 0
```

```
txt_status.Text = txt_status.Text + Chr$(13) + Chr$(10) + "No se ha  
encontrado el Telefono en el archivo de configuracion."
```

```
Exit Function
```

```
FinMTX:
```

```
Me.MousePointer = 0
```

```
End Function
```

```
RECIBIENDO DATOS
```

```
Sub RecibiendoDatos ()
```

```
Dim resp As String, alarma$, celda$, estado$, mensaje$, c$
```

```
Dim pos As Long
```

```
comm_puerto.InputLen = 1
```

```
Do
```

```
    If Cerrar Then Exit Do
```

```
    If comm_puerto.InBufferCount > 0 Then
```

```
        linea = 0
```

```
    Leer:
```

```
        pos = 0
```

```
resp = ""
Do
    c = comm_puerto.Input
    resp = resp + c
    DoEvents
    If Cerrar Then Exit Sub
Loop While c <> ">"
txt_status.Text = txt_status.Text + resp
If linea = 0 Then
    pos = InStr(pos + 1, resp, "MTX")
    If pos > 0 Then
        alarma = Mid$(resp, pos + 3, 3)
        linea = 1
        GoTo Leer
    Else
        alarma = ""
    End If
End If
If linea = 1 Then
    pos = InStr(pos + 1, resp, "CELL =")
    If pos > 0 Then
        celda = Mid$(resp, pos + 7, 3)
        linea = 2
        pos = InStr(pos + 1, resp, "ALARM =")
        If pos > 0 Then
            mensaje = Trim$(Mid$(resp, pos + 7, 27))
        Else
            mensaje = ""
        End If
    End If
```

```

        GoTo Leer
    Else
        celda = ""
    End If
End If
If linea = 2 Then
    pos = InStr(pos + 1, resp, "NEW STATE:")
    If pos > 0 Then
        estado = Trim$(Mid$(resp, pos + 12, 6))
        linea = 3
    Else
        estado = ""
    End If
End If
If linea = 3 Then
    If (alarma <> "" Or celda <> "") Then
        If UCase$(estado) = "ACTIVE" Then
            fx_add_alarma alarma, celda, mensaje
        End If
    End If
End If
Else
    AtenderMensajes
End If
DoEvents
Loop While True
End Sub
SETEAR ALARMA
Sub SetearAlarma (alarma As String, celda As String)

```

```

lbl_alarma.Caption = alarma
lbl_celda.Caption = celda
pnl_alarma.Visible = True
tmr_msg.Enabled = True
End Sub
DECLARACION DE VARIABLES
Global db_beeper As database
Global DatabaseName$
Global Const IniFile$ = "mtx.ini"
Global mybase As database
Declare Function GetPrivateProfileString Lib "Kernel" (ByVal
IpApplicationName As String, ByVal IpKeyName As Any, ByVal IpDefault As
String, ByVal IpReturnedString As String, ByVal nSize As Integer, ByVal
IpFileName As String) As Integer
'-----
'Comm Control
'-----
'Handshaking
Global Const MSCOMM_HANDSHAKE_NONE = 0
Global Const MSCOMM_HANDSHAKE_XONXOFF = 1
Global Const MSCOMM_HANDSHAKE_RTS = 2
Global Const MSCOMM_HANDSHAKE_RTSXONXOFF = 3
'Event constants
Global Const MSCOMM_EV_SEND = 1
Global Const MSCOMM_EV_RECEIVE = 2
Global Const MSCOMM_EV_CTS = 3
Global Const MSCOMM_EV_DSR = 4
Global Const MSCOMM_EV_CD = 5
Global Const MSCOMM_EV_RING = 6

```

Global Const MSCOMM\_EV\_EOF = 7

'Error code constants

Global Const MSCOMM\_ER\_BREAK = 1001

Global Const MSCOMM\_ER\_CTSTO = 1002

Global Const MSCOMM\_ER\_DSRTO = 1003

Global Const MSCOMM\_ER\_FRAME = 1004

Global Const MSCOMM\_ER\_OVERRUN = 1006

Global Const MSCOMM\_ER\_CDTO = 1007

Global Const MSCOMM\_ER\_RXOVER = 1008

Global Const MSCOMM\_ER\_RXPARITY = 1009

Global Const MSCOMM\_ER\_TXFULL = 1010

CONECTARDB

Function ConectarDB () As Integer

    Dim field\$, ret%

    On Error Resume Next

    ConectarDB = False

    field\$ = Space(129)

    ret = GetPrivateProfileString("General", "Base", "", field\$, 128, IniFile\$)

    If ret < 0 Then

        MsgBox "Base de datos no ha sido especificada", 16, "Coneccion a la Base de Datos"

        Exit Function

    Else

        field\$ = Trim(Left(field\$, ret))

    End If

    DatabaseName = field\$

    Set db\_beeper = OpenDatabase(DatabaseName, False, False, "")

    If Err <> 0 Then

        MsgBox Error, 16, "Coneccion a la Base de Datos"

**Exit Function**

**End If**

**ConectarDB = True**

**End Function**

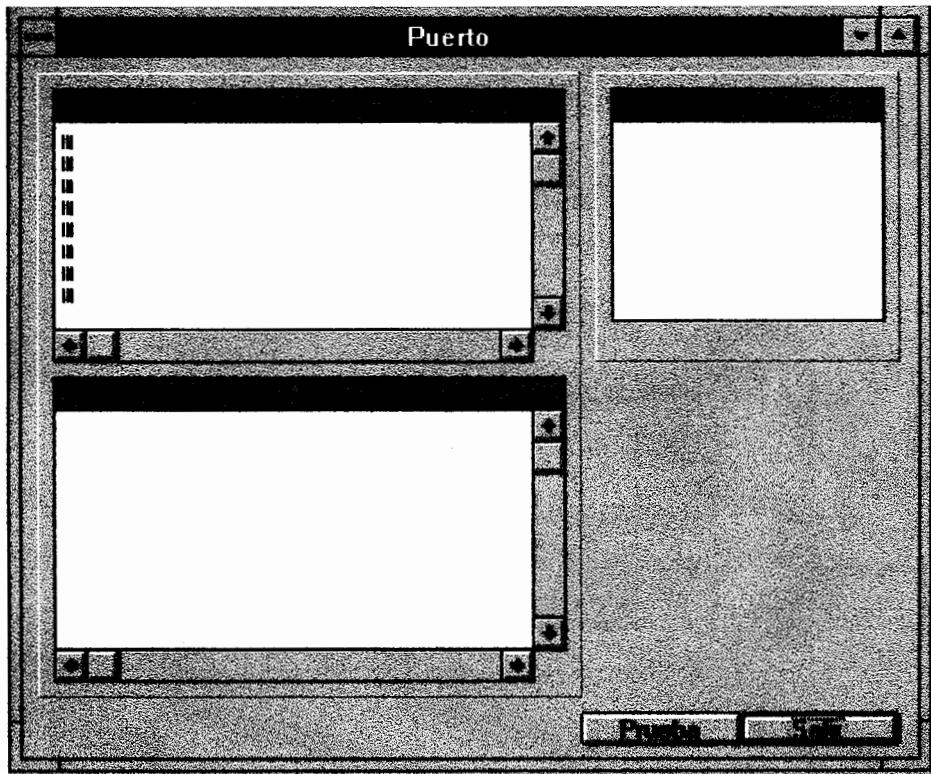


**ANEXO**

---

**MANUAL DE USUARIO**

# MONITOREO DE ALARMA



A esta pantalla se llega luego de dar click en “Monitoreo de Alarmas” de la PANTALLA PRINCIPAL.

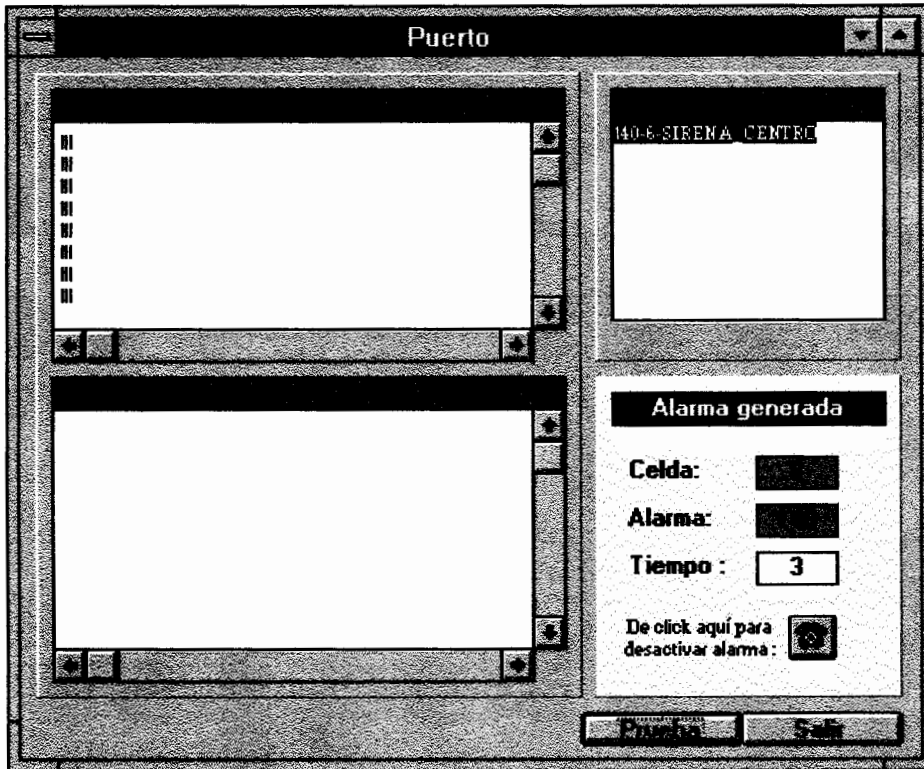
Esta pantalla consta de las siguientes partes :

1. Cuatro secciones de Monitoreo
  - 1.1 Log Puerto MTX
  - 1.2 Log Puerto Beeper
  - 1.3 Cola de Alarmas
  - 1.4 Alarma Generada (Activada en el momento de enviar la Alarma)
2. Botón Prueba
3. Botón Salir



**Biblioteca Central**

## ACTIVACION DE ALARMA



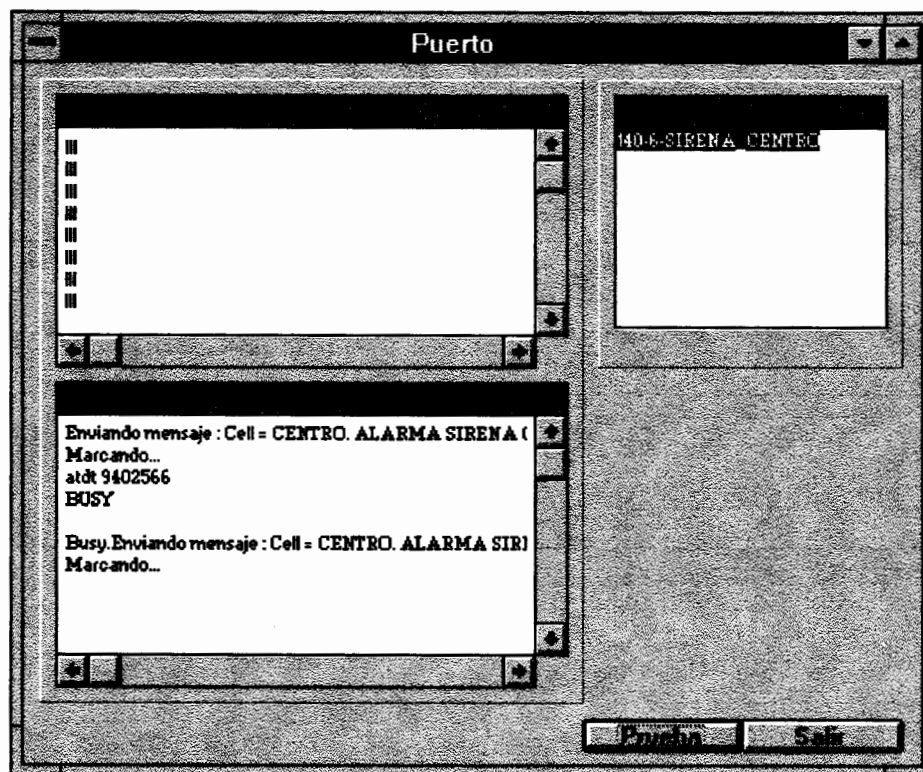
Una vez que se pone en marcha el programa, ejecutándolo desde el archivo ALARMAS.EXE, el programa automáticamente comienza a monitorear el puerto del MTX en espera de la activación de alguna alarma que halla sido enviada desde cualquiera de las celdas que al momento constan en CONECCELL. Este monitoreo se lo visualiza en el recuadro "Log Puerto MTX".

En el recuadro ALARMA GENERADA, se muestra la celda y el código de la alarma que se activó y esta lista para ser enviada al Beeper. En caso de querer detener el envío de dicha alarma, se debe dar click en el botón del Teléfono.

En el recuadro COLA DE ALARMAS, se presenta la lista de alarmas en caso

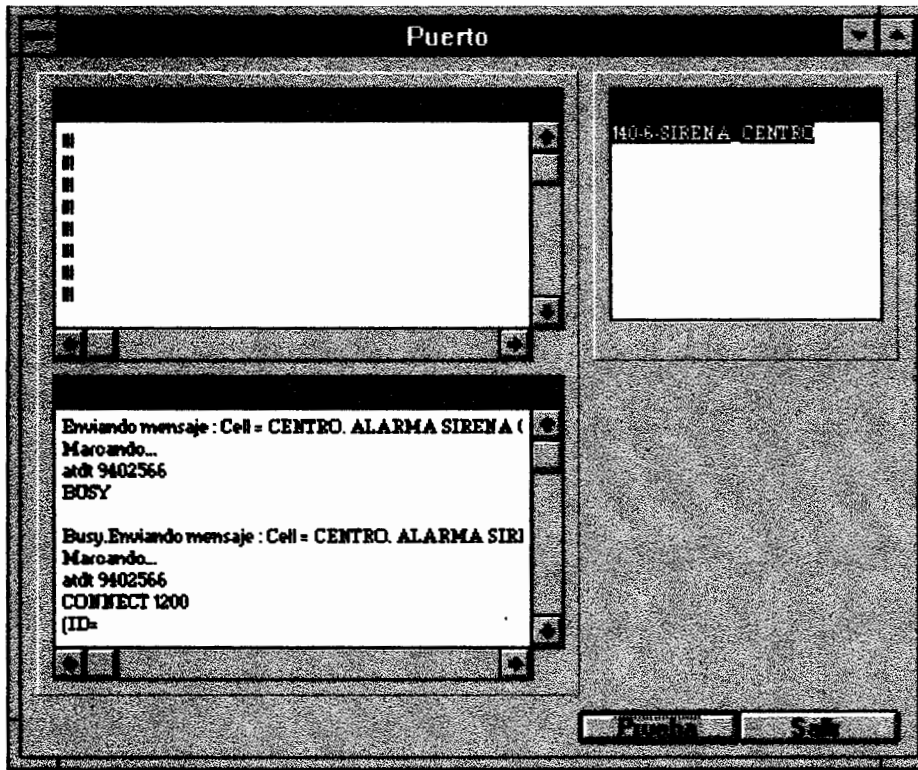
de ser varias, que serán enviadas de forma irrevocable al Beeper. Cabe mencionar que la primera alarma de la lista es la primera en ser enviada.

## COMUNICACIÓN CON LA CENTRAL DE BEEPER



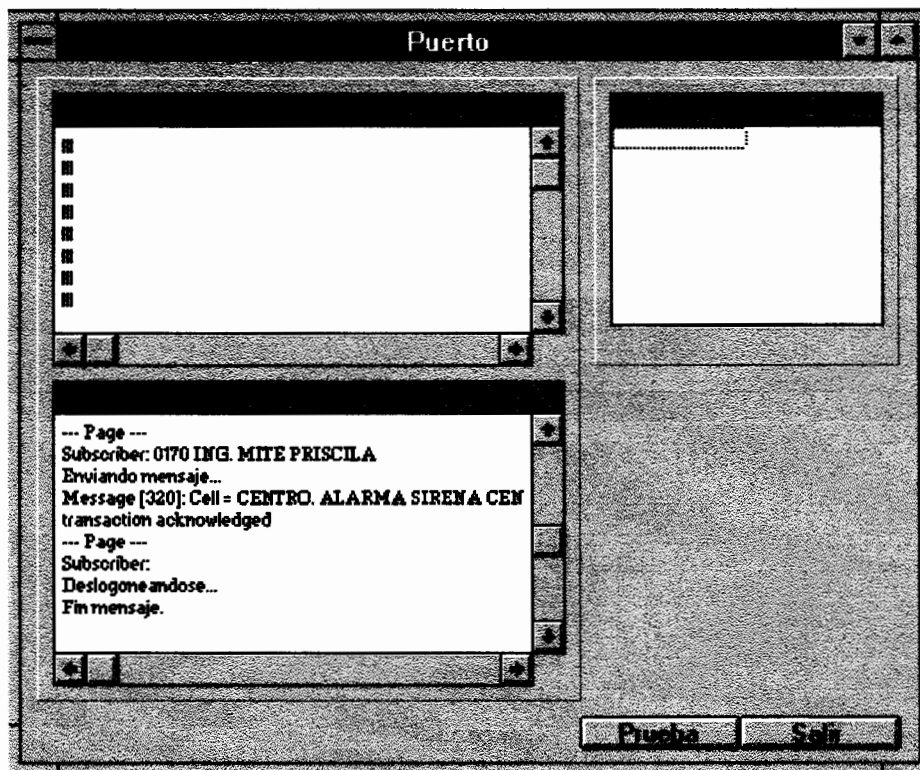
En el recuadro "Log Puerto Beeper" aparece la Alarma que va a ser enviada a la Central de Beeper, para lo cual el sistema marca automáticamente el número telefónico de la Central (9402566), en caso de que la línea esté ocupada, aparece la palabra BUSY. El sistema reintentará la comunicación con la Central cuantas veces sea necesario, hasta lograr la comunicación.

## CONEXION CON LA CENTRAL BEEPER



Una vez lograda la comunicación con la Central Beeper, en el recuadro de "Log Puerto Beeper". Aparece *CONNECT 1200*, lo cual indica que la conexión a la Central ha sido exitosa.

## ENVIO DEL MENSAJE AL BEEPER



En esta pantalla se observa el número 0170 que es el código del Beeper, el Nombre de la persona a quien se le está enviando el mensaje y el mensaje que para este caso es "ALARMA SIRENA CENTRO" . Una vez que el mensaje a sido recibido por el Beeper, la Central envía el mensaje TRANSACTION ACKNOWLEDGED, con lo que se finaliza la comunicación.

## BIBLIOGRAFIA

- Guía completa de **Visual Basic** para Windows  
Segunda Edición. Por Ross Nelson. Editorial Mc Graw Hill.
- **Cell Site Manual NT-800DR (ICP)**  
De Northern Telecom.
- **ZETRON**  
Operating & programming manual. Version 310.  
De Copyright Zetron, Inc.



**Biblioteca Central**