



ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y
COMPUTACION

“DISEÑO E IMPLEMENTACION EN LA CENTRAL TRANSITO
GUAYAQUIL DE UN ADMINISTRADOR DE MEDICION DE
TRAFICO Y CALIDAD DE SERVICIO CON CENTRALES
ERICSSON DE LA RED METROPOLITANA DE GUAYAQUIL”

PROYECTO DEL TOPICO DE GRADUACION
“CONMUTACION TELEFONICA”

PREVIO A LA OBTENCION DEL TITULO DE:
INGENIERO EN ELECTRICIDAD ESPECIALIZACION
ELECTRONICA - TELECOMUNICACIONES

PRESENTADO POR:

VICTOR HUGO BEDOYA PILOZO
HENRY GABRIEL GOMEZ BASILIO
LUIS ENRIQUE PITA MATEO
ROMMY ESPERANZA TORRES MARRIOTT

GUAYAQUIL - ECUADOR
1997

INDICE GENERAL

CONTENIDO	PAG.
INTRODUCCION	
CAPITULO 1	
MEDICIONES EN EL SISTEMA AXE	
1.1 DESCRIPCION DEL SISTEMA AXE	9
1.2 MEDICIONES DE TRÁFICO	10
1.3 MEDICIONES DE TRÁFICO EN HORA PICO	11
1.4 MEDICIONES DE TIPO DE TRÁFICO	12
1.5 MEDICIONES DE CARGA DEL PROCESADOR	13
1.6 MEDICIONES DE CALIDAD DE SERVICIO	13
CAPITULO 2	
ANALISIS DEL SISTEMA ACTUAL	
2.1 DESCRIPCION, EJECUCION E INTERPRETACION DE COMANDOS	15
2.1.1 COMANDOS DE MEDICION DE TRÁFICO	15
2.1.2 COMANDOS DE MEDICION DE TIPO DE TRÁFICO	20
2.1.3 COMANDOS DE MEDICION DE CARGA DE PROCESADOR.	23
2.1.4 COMANDOS DE MEDICION DE CALIDAD DE SERVICIO.	25

CAPITULO 3

DISEÑO E IMPLEMENTACION DEL ADMINISTRADOR

3.1 BENEFICIOS DEL ADMINISTRADOR	30
3.2 ESTRUCTURA DE FUNCIONAMIENTO	31
3.2.1 DIAGRAMA DE NIVEL 0	32
3.2.2 DIAGRAMA DE NIVEL 1 PARA CALIDAD	33
3.2.3 DIAGRAMA DE NIVEL 1 PARA TRAFICO	34
3.2.4 DIAGRAMA DE NIVEL 1 PARA CARGA	35
3.2.5 DIAGRAMA DE NIVEL 2 PARA CALIDAD	36
3.2.6 DIAGRAMA DE NIVEL 2 PARA TRAFICO	37
3.3 CODIGO FUENTE	38
3.3.1 DECLARACION DE VARIABLES GLOBALES.	38
3.3.2 LIMPIAR TEXTO	38
3.3.3 TRANSFERENCIA DE TEXTO	38
3.3.4 VALIDA TEXTO	39
3.3.5 LEE UN PARAMETRO	40
3.3.6 REEMPLAZA STRING	41
3.3.7 CALCULO DE FACTORIAL	41
3.3.8 CALCULO DE NUMERO DE CIRCUITOS	42
3.3.9 PROCESO DE SUMATORIAS	42
3.3.10 PROCESAR CALIDAD	43
3.3.11 PROCESAR CARGA	44
3.3.12 PROCESAR PICO	46
3.3.13 PROCESAR RUTAS	48
3.3.14 CONECTAR BASE	50
3.3.15 RECIBIR DATOS	51
3.3.16 TABLA TEMPORAL	51

CAPITULO 4	
OPERACION DEL ADMINISTRADOR Y RESULTADOS	
4.1 DESCRIPCION GENERAL	52
4.2 MULTI-SELECCION PARA MEDIR CALIDAD DE SERVICIO	53
4.3 MULTI-SELECCION PARA MEDIR TRAFICO	59
4.4 MULTI-SELECCION PARA MEDIR CARGA DEL PROCESADOR	68
4.5 ADMINISTRADOR DE LA BASE DE DATOS	73
4.5.1 VENTANA DE PRESENTACION	74
4.5.2 SELECCION DE BASE DE CENTRAL	74
4.5.3 SELECCION DE BASE DE RUTAS	76
4.5.4 SELECCION DE BASE DE GRUPO DE CENTRALES	78
4.5.5 ACTUALIZACION DE BASE DE DATOS	80
CONCLUSIONES Y RECOMENDACIONES	82
BIBLIOGRAFIA	85
ANEXOS	
A. CARACTERISTICAS DE HARDWARE Y SOFTWARE	86
PROCESO DE INSTALACION DEL PROGRAMA	87
RESTRICCIONES DEL PROGRAMA	87
B. DEFINICIONES GENERALES	89
CONCEPTO DE TRAFICO TELEFONICO	90
C. CODIGOS DE SUBROUTINAS	98

AGRADECIMIENTO

En primer lugar agradecemos a Dios y a nuestros padres por habernos dado la vida y la capacidad necesaria para llegar hasta el presente t3pico.

Para poder realizar el presente trabajo tuvimos que tocar muchas puertas, de las cuales muy pocas fueron abiertas. Es por esto que no podriamos dejar de nombrar a aquellas personas que nos ayudaron desinteresadamente.

Una menci3n especial merece el Ing. Vicente Saltos B. ya que gracias a su vasta experiencia en el campo de las telecomunicaciones as3 como a sus excelentes relaciones, pudimos llevar a cabo nuestro proyecto.

El principal problema con el que nos encontramos fue la obtenci3n de informaci3n, pero gracias a personas tales como:

Tecn. Carlos Zambrano

Ing. Patricia Abad,

es que se consigui3 una visi3n general de todos los procesos y de los informes finales a los que debiamos llegar.

Tambi3n agradecemos a todas las personas que de una u otra forma contribuyeron con la elaboraci3n de los programas de procesamiento de datos en los diversos lenguajes que se usaron.

De la misma forma terminamos agradeciendo a todas las personas que nos ayudaron con diversos equipos de computaci3n para desarrollar los programas antes mencionados.

DEDICATORIA

Dedicamos el presente trabajo a nuestros padres y a las personas que de una u otra forma nos dieron el empuje necesario para poder culminarlo.

DECLARACION EXPRESA

"La responsabilidad de los hechos, ideas y doctrinas expuestos en este informe técnico, nos corresponden **exclusivamente**, y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL"

VICTOR HUGO BEDOYA PILOZO



HENRY GABRIEL GOMEZ BASILIO



LUIS ENRIQUE PITA MATEO

ROMMY ESPERANZA TORRES MARRIOTT





Ing. Raúl Noriega



Dr. Freddy Villao Quezada



Ing. Washington Medina Moreira

RESUMEN

CAPITULO 1

En este capítulo se realiza una breve descripción del sistema AXE que es el que se encarga de obtener las mediciones de las centrales ERICSSON tales como: mediciones de calidad de servicio, tráfico, tráfico en hora pico, tipos de tráfico y carga del procesador.

CAPITULO 2

En el capítulo 2 se realiza un análisis de la operación del sistema de mediciones actual, en la que se detallan los comando que se usan para la medición de tráfico por rutas, de tipo de tráfico, de carga del procesador y de calidad de servicio. Los datos que se obtienen en estas mediciones son los que sirven para obtener una visión real del funcionamiento de las centrales.

CAPITULO 3

Aquí se muestra el diseño e implementación del Administrador de las mediciones de tráfico, calidad de servicio y de carga del procesador. Primero se indican los beneficios de nuestro programa, luego se muestra la estructura de funcionamiento del mismo por medio de diagramas de niveles y por último se incluyen los códigos fuente de los principales procesos del programa.

CAPITULO 4

En el presente capítulo se hace una presentación en forma gráfica de las principales pantallas que genera nuestro Administrador de Medición de Tráfico y

Calidad de Servicio. Dicha presentación viene acompañada de una breve explicación de sus respectivas funciones, sirviendo de guía para el operador.

ANEXO A

Es un manual para la instalación y operación del programa, dando las características correspondientes del hardware y del software.

ANEXO B

Anexamos diversos conceptos importantes de los diferentes términos utilizados en la telefonía.

ANEXO C

Diversas subrutinas que ayudan al estudio y entendimiento de nuestro diseño.

INTRODUCCION

Para obtener un servicio telefónico rápido y eficaz se requiere de una operación apropiada y por supuesto de un excelente dimensionamiento para el mismo. Siendo así que actualmente el procedimiento para obtención de circuitos y gráficas estadísticas de tráfico y calidad de servicio se realiza de una manera semiautomática, en la cual se "bajan" datos del sistema para luego en base a ellos realizar cálculos que después son convertidos en parámetros útiles tales como: circuitos, erlangs, congestionamiento, llamadas procesadas, etc. y luego son transformados en porcentajes estadísticos de manera gráfica.

El presente administrador es un conjunto de programas que en intercomunicación forman un proceso que provee de un excelente servicio de control de centrales telefónicas. Este trabajo consta de los programas administradores además de los programas-herramientas que tiene el sistema.

El trabajo representa un significativo esfuerzo en el intento a proveer de una referencia rápida, sea un sistema de gran capacidad y flexibilidad. Ofreciendo a los destinatarios el conocimiento de como funcionan las herramientas de modo que puedan utilizarlas.

CAPITULO 1

MEDICIONES EN EL SISTEMA AXE

1.1 DESCRIPCION DEL SISTEMA AXE

La central Ericsson utiliza el sistema AXE, por tal motivo analizaremos este sistema a continuación:

Sistema Axe.-

AXE es un sistema SPC "Stored Program Central" lo que significa programa almacenado en una computadora que controlan la operación de la central. Todas las operaciones son almacenadas en una memoria.

Para cambiar algo en AXE debemos cambiar las instrucciones en la memoria, El equipo de conmutación esta controlado por dos computadoras uno en estado de espera.

Este sistema se divide en dos grandes grupos: APT que son los equipos de conmutación y el APZ que es el software, la computadora.

El APZ consta de dos procesadores, uno hace el barrido rutinario para detectar cambios y reporta los eventos importantes al CP se lo conoce como RP. CP hace los análisis y diagnósticos que requieren alta capacidad de computación.

Tipos de Centrales existentes.-

En el Ecuador existen varios tipos de centrales entre ellas:

- Centrales Digitales: Ericsson (AXE-10), Alcatel (E10B) y las Siemens (EWSD).
- Centrales Analógicas: ARF 102

1.2 MEDICIONES DE TRAFICO

Las mediciones de tráfico en rutas implica el registro del tráfico en las distintas rutas de una central, sean con el sistema de pérdida o de cola (cada tipo proporciona un resultado distinto).

Las estadísticas registradas en este tipo de medición proporcionan información detallada sobre el comportamiento de las rutas de una central y son útiles para la planificación de la red.

La medición proporciona estadísticas que incluyen lo siguiente:

- * La intensidad de tráfico en Erlangs en una ruta
- * El número de intentos de ocupación de circuitos
- * El porcentaje de congestión debido a falta de circuitos libres en una ruta.
- * El número de circuitos pertenecientes en una ruta.
- * Número promedio de circuitos bloqueados en una ruta.
- * Tiempo promedio de ocupación en segundos de los circuitos de una ruta.
- * Número de llamadas contestadas en una ruta.
- * Número de intentos en cola (cuando la ruta tiene sistema de colas).

- * Número de ocupaciones después de la cola (cuando la ruta tiene sistema de colas).
- * Promedio de circuitos en la cola (cuando la ruta tiene sistema de colas).
- * Tiempo en cola de un circuito (cuando la ruta tiene sistema de colas).

1.3 MEDICIONES DE TRÁFICO EN HORA PICO

En los sistemas de telefonía existe un parámetro que es tomado muy en cuenta por los operadores de las centrales, el cual es la determinación del tráfico en la hora pico.

En el sistema actual de EMETEL, existe un programa que realiza ese cálculo de la hora pico, el mismo que nos arroja los siguientes resultados:

- * Hora pico.
- * Ruta analizada.
- * Valor del tráfico en esa ruta a esa hora pico.
- * Intentos de ocupación de circuitos en esa hora.
- * Congestión a esa hora.
- * Número total de circuitos actuales en esa red.
- * Número de contestaciones de B.
- * Tiempo de ocupación en segundos

Esta medición nos indica la hora del día en la cual se ha registrado el mayor tráfico y nos sirve para determinar si el sistema logra abastecer correctamente este flujo de llamadas o si existe congestión.

En base al valor de Erlangs que se saca en esta hora pico se calculan el número de circuitos ideales que se necesitarían para abastecer normalmente un cierto porcentaje de grado de servicio.

Luego se debe comparar si es que la cantidad de circuitos que se ha calculado corresponde a los circuitos existentes en la realidad.

En el caso de que el valor calculado no corresponda con la realidad, dependerá del análisis comercial que se haga para poder saber si conviene o no realizar una expansión de los circuitos.

1.4 MEDICIONES DE TIPO DE TRÁFICO

La medición de tipo de tráfico implica la medición de todos los tipos existentes en una central AXE.

Los tipos de tráfico que se pueden medir son:

- * Tráfico entrante
- * Tráfico saliente
- * Tráfico originado
- * Tráfico terminal
- * Tráfico en tránsito
- * Tráfico interno

La medición proporciona estadística por tipo de tráfico que incluye lo siguiente:

- * La intensidad de tráfico en Erlangs.
- * Número de intentos de ocupación de circuitos.
- * Porcentaje de llamadas infructuosas.
- * Número total de circuitos unidireccionales.
- * Número total de circuitos bidireccionales.
- * Número total de circuitos unidireccionales bloqueados.
- * Número total de circuitos bidireccionales bloqueados.

1.5 MEDICION DE CARGA DE PROCESADOR

La carga del procesador es un término que se usa para describir la capacidad del procesador que se está usando .

La medición de carga del procesador proporciona una lectura referente a la carga del procesador junto con datos sobre la cantidad de llamadas originadas y entrantes por segundos que se han intentado y procesado.

Razones por la cual se realiza la medición de carga del procesador es para determinar si la central requiere un procesador más grande y para determinar los tiempos de vaciado.

La medición proporciona estadísticas que incluye lo siguiente:

- + Porcentaje de carga del procesador.
- + Promedio de llamadas originadas/seg. tanto las que se han intentado como las procesadas.
- + Promedio de llamadas entrantes/seg. tanto las que se han intentado como las procesadas.

1.6 MEDICIONES DE CALIDAD DE SERVICIO

El propósito de las estadísticas de calidad de servicio es medir la calidad de servicio en distintos casos de tráfico.

Se miden las estadísticas de calidad de servicio en base a muestras aleatorias de tráfico real durante un periodo de tiempo.

Las estadísticas de calidad de servicio registran datos sobre:

- * Llamadas exitosas.
- * Razón de las llamadas fracasadas.
- * Ciertas mediciones de tiempo.

Las llamadas registradas se dividen según eventos de interés por lo que la medición proporciona :

- * Llamada con marcación incompleta.
- * Llamadas donde A repone.
- * Llamadas hacia números B libres.
- * Llamadas hacia números B que no contestaron.
- * Llamadas hacia números B que contestaron.
- * Llamadas hacia números B ocupados.
- * Llamadas hacia números B inobtenible.
- * Llamadas que encontraron congestión dentro de la central.
- * Llamadas que encontraron congestión fuera de la central.
- * Llamadas fracasadas por fallas técnicas en la central o en transmisión.
- * Tiempo promedio de conversación.
- * Tiempo promedio que transcurre hasta la contestación de B

CAPITULO 2

ANALISIS DEL SISTEMA ACTUAL

2.1 DESCRIPCION, EJECUCION E INTERPRETACION DE COMANDOS

Actualmente el sistema de mediciones no es aprovechado al máximo , debido a que en EMETEL no se realiza un total análisis de todos los datos generados por las diferentes centrales (en nuestro caso las centrales AXE).

A continuación se detalla los diferentes comandos:

2.1.1. COMANDOS DE MEDICION DE TRÁFICO POR RUTAS

A.- Comandos e Impresos

Comandos:

TRRGI	Medición de tráfico en rutas, iniciación del grupo de registro.
TRRGE	Medición de tráfico en rutas, borrado del grupo de registro.
TRRGP	Medición de tráfico en rutas, impresión del grupo de registro.
TRRPI	Medición de tráfico en rutas, iniciación del programa de medición.
TRRPP	Medición de tráfico en rutas, impresión del programa de medición.
TRRIC	Medición de tráfico en rutas, cambio del direccionamiento de impresión.
TRIDP	Medición de tráfico en rutas, identidad del próximo programa de medición libre.
TRTSE	Registro de tráfico, finalización del programa de tiempo.

TRTSI Registro de tráfico, iniciación del programa de tiempo.
 TRTSP Registro de tráfico, impresión del programa de tiempo.

Impresos:

TRAFFIC MEASUREMENT ON ROUTES GROUP DATA
 TRAFFIC MEASUREMENT ON ROUTES MEASURING PROGRAM
 TRAFFIC MEASUREMENT ON ROUTES RESULT, LSR
 TRAFFIC MEASUREMENT ON ROUTES RESULT, QSR
 TRAFFIC RECORDING TIME SCHEDULE
 TRAFFIC RECORDING IDENTITY

B.- Ejecución

El procedimiento es igual para rutas con sistema de pérdida o cola.

a.- Definir el grupo de registro.

<TRRGI; TRG = 8, R=STDO&STDI&CCLO&CCLI&IQ40&IQ41;

(R máximo puede ser hasta 32 rutas del mismo tipo)

b.- Imprimir el contenido de los grupos de registro.

<TRRGP TRG=trg;

TRAFFIC MEASUREMENT ON ROUTES GROUP DATA

TRG	SI	R1	R2	SIB	PXR
8	60	STDO			
		STDI			
		CCLO			
		CCLI			
		IQ40			
		IQ41			

NRTG:

120

END

c.- Solicitar un programa de medida libre:

```
<TRIDP,
TRAFFIC RECORDING IDENTITY
MP=
11
END
```

d.- Definir el programa de medición para la medición de tráfico en rutas.

```
<TRRPI MP=11,TRG=8.
```

e.- Imprimir los datos del programa de medición.

```
<TRRPP MP=ALL
TRAFFIC MEASUREMENT ON ROUTES MEASURING PROGRAM
MP      STATE      TRG      IO      FILE
11      DEFINED      8              AT 0
END
```

(si FILE o IO no son especificados, por defecto queda definido para salida de resultados el terminal desde el que se envió el comando).

f.- Definir el programa de tiempo para el programa de medición

```
<TRTSI MP=11,TIME=0930,RPL=60,NDAYS=365
```

g.- Imprimir los valores del programa de tiempo

```
<TRTSP MP=11
TRAFFIC RECORDING TIME SCHEDULE
MP  NRP  RPL  DATENDAYSDCAT  TIME
11  1    60  950701    365  ALL    0930
END
```

(algunos valores son especificados por defecto, ver la interpretación de resultados).

C.- Interpretación de resultados

RESULTADOS PARA RUTAS CON SISTEMA DE PERDIDA

RESULTADO:

TRAFFIC MEASUREMENT ON ROUTES RESULT, LSR

TRG	MP	NRP	RPL	GRN	DATE	TIME	SI	NM	FCODESE
8	11	1	60	1	950701	0000	60	NO	
R	TRAFFIC	NBIDS	CCONG	IDV	ANBLO	MHTIME	NBANSW		
STDD	28.7	1691	0.0	339	0.0	74.3	569		
STDI	31.2	1691	339		0.0	68.4	599		
CCLD	24.5	1750	27.8	30	0.0	69.8	452		
CCLI	3.3	135		30	0.0	88.0	54		
IQ40	5.1	135	0.0	123	0.0	92.2	53		
IQ41	3.2	125	123		0.0	136.8	78		
END									

PARAMETROS:

TRG	Identidad de grupo de registro de tráfico.
MP	Identidad del programa de medición.
NRP	Numero de veces al dia que se efectuara la medición.
RPL	Duración en minutos del tiempo de medición.
RPN	Numero de la medición en el dia.
GRN	Numero de la medición desde que fue definida.
DATE	Fecha de inicio de la medición (AAMMDD)
TIME	Hora de inicio de la medición (HHMM)
SI	Intervalo para la exploración de contadores en segundos.
NM	Indica si la función de gestión de red esta en operación.
FCODE	Código de falla.
TRAFF	Intensidad de tráfico en erlangs.
NBIDS	Intentos de llamadas.

CCONG	Porcentaje de congestión por falta de circuitos (solo rutas salientes).
NDV	Numero de circuitos que estan conectados a una ruta.
ANBLO	Promedio de circuitos bloqueados en una ruta.
MHTIME	Tiempo de ocupación en segundos.
NBANSW	Numero de contestaciones de B.

RESULTADO PARA RUTAS CON SISTEMA DE COLA.

RESULTADO:

TRAFFIC MEASUREMENT ON ROUTES RESULT, LSR

TRG	MP	NRP	RPL	GRN	DATE	TIME	SI	NM	FCODESF		
8	11	1	60	1	950701	0930	60	NO			
R		TRAFF	NBIDS	NBIDSQ	NSAQ	AQI	NDV	ANBLO	MHTIME	MQTIME	
CSRR2S	23	220	0	0	0.0	32	0	3.1	0.0		
CSRR2R	33	262	0	0	0.0	32	0	1.8	0.0		

PARAMETROS:

TRG	Identidad de grupo de registro de tráfico
MP	Identidad del programa de medición
NRP	Numero de veces al dia que se efectuara la medición
RPL	Duración en minutos del tiempo de medición
RPN	Numero de la medición en el dia
GRN	Numero de la medición desde que fue definida
DATE	Fecha de inicio de la medición (AAMMDD)
TIME	Hora de inicio de la medición (HHMM)
SI	Intervalo para la exploración de contadores en segundos
NM	Indica si la función de gestión de red esta en operación
FCODE	Código de falta
TRAFF	Intensidad de tráfico en erlangs
NBIDS	Intentos de llamadas

NBIDSQ	Número de intentos que entraron en cola
NSAQ	Número de ocupaciones después de la cola
AQL	Número de circuitos en cola
NDV	Número de circuitos conectados
ANBLO	Número de contestaciones de B.
MHTIME	Tiempo de ocupación en segundos
MQTIME	Tiempo en cola en segundos

2.1.2 COMANDOS DE MEDICION DE TIPO DE TRÁFICO

A.- Comandos e Impresos

Comandos:

TRIDP	Identidades de grabación de datos, imprimir
TRATI	Medición de tipo de tráfico, iniciación del programa de medida
TRATC	Medición de tipo de tráfico, cambio de direccionamiento de impresión.
TRTSE	Tabla de tiempo de grabación de tráfico, finalización
TRTSI	Tabla de tiempo de grabación de tráfico, inicio
TRTSP	Tabla de tiempo de grabación de tráfico, imprimir

Impresos:

TRAFFIC RECORDING IDENTITY
 TRAFFIC RECORDING TIME SCHEDULE
 TRAFFIC TYPE MEASUREMENT PROGRAM
 TRAFFIC TYPE MEASUREMENT RESULT

B.- Ejecución:

a.- Solicitar un programa de medida libre

<TRIDP.

TRAFFIC RECORDING IDENTITY

```
MP
12
END
```

b.- Definir el programa de medición para la medición de tipo de tráfico.

```
<TRATI MP=12
```

c.- Imprimir los datos de los programas de medición para la medición de tipo de tráfico.

```
<TRRPP MP=ALL
```

TRAFFIC TYPE MEASUREMENT PROGRAM

```
MP          STATE          IO          FILE
12          DEFINED        AT 0
END
```

(si FILE o IO no son especificados, por defecto queda definido para salida de resultados el terminal desde donde se envió el comando)

d.- Definir el programa de tiempo para el programa de medición.

```
<TRTSI MP=12 TIME=0000 RPL=60 NDAYS=3 NRP=24 DATE=950702
```

e.- Imprimir los valores del programa de tiempo

```
<TRTSP MP=12
```

TRAFFIC RECORDING TIME SCHEDULE

```
MP   NRP   RPL   DATE   NDAYS   DCAT   TIME
12   24    60    950702  3       ALL    0000
END
```

(algunos valores son especificados por defecto, ver la interpretación de resultados)

C.- Interpretación de Resultados.

RESULTADO:

TRAFFIC TYPE MEASUREMENT RESULT							
MP	NRP	RPL	RPN	GRN	DATE	TIME	NM
12	24	60	1	1	950702	0000	NO
TYPE	TRAFF	NBIDS	UNSUC	NDVO	NDVB	NBLOO	NBLOB
ORG	0.0	0	0.0	0	0	0.0	0.0
IEX	2645.3	120896	1.0	0	10721	0.0	77.8
TE	0.0	0	0.0	0	0	0.0	0.0
OEX	2643.4	119773	0.8	0	10721	0.0	77.8
INT	0.0	0	0.0				
TRA	2648.4	119773	0.8				
END							

PARAMETROS:

MP	Identidad del programa de medición.
NRP	Numero de veces al dia que se efectuara la medición.
RPL	Duración en minutos del tiempo de medición.
RPN	Numero de la medición en el dia.
GRN	Numero de la medición desde que fue definida.
DATE	Fecha de inicio de la medición (AAMMDD).
TIME	Hora de inicio de la medición (HHMM).
NM	Indica si la función de gestión de red esta activa.
TYPE	Tipo de grafico ; ORG,IEX,TE,OEX,INT,TRA.
TRAFF	Intensidad de tráfico en erlangs.
NBIDS	Intentos de ocupación de circuitos.
UNSUC	Porcentaje de llamadas infructuosas.
NDVO	Numero total de circuitos unidireccionales.
NDVB	Numero total de circuitos bidireccionales.
NBLOO	Promedio de circuitos unidireccionales bloqueados.
NBLOB	Promedio de circuitos bidireccionales bloqueados.

2.1.3 COMANDOS DE MEDICIÓN DE CARGA DEL PROCESADOR

A.- Comandos e impresos:

Descripción de comandos:

PLSMI	Medición de carga del procesador, iniciación del programa de medida.
PLSMP	Medición de carga del procesador, impresión del programa de medida
TRIDP	Identidades de grabación de datos, imprimir.
TRTSE	Tabla de tiempo de grabación de tráfico, finalizar.
TRTSI	Tablas de tiempo de grabación de tráfico, iniciar.
TRTSP	Tabla de tiempo de grabación de tráfico, imprimir.

Descripción de mensajes impresos:

TRAFFIC RECORDING IDENTITY
 PROSESSOR LOAD MEASUREMENT PROGRAM
 TRAFFIC RECORDING TIME SCHEDULE

B.- Ejecución.

a.- Solicitar un programa de medida libre:

```
<TRIDP
TRAFFIC RECORDING IDENTITY
MP
13
END
```

b.- Definir el programa de medición para la medición de carga del procesador

```
<PLSMI MP=13
```

c.- Imprimir los datos de los programas de medición para la medición de carga del procesador.

```
<PLSMP MP=ALL
```

PROCESSOR LOAD MEASUREMENT PROGRAM

```
MP IO STATE
13 AT-0 DEFINED
END
```

(si IO no es especificado, por defecto quedara definido para salida de resultados el terminal desde donde se enviò el comando).

C.- Interpretación de resultados

RESULTADO:

PROCESSOR LOAD MEASUREMENT RESULT

MP	NRP	RPL	RPER	GRN	DATE	TIME
13	24	60	1	1	950702	0000
PLOAD	OFFDO	OFFDI	FTCHDO	FTCHDI		
9	0.0	6.5	0.0	0.5		

PARAMETROS:

MP	Identidad del programa de medición.
NRP	Numero de veces al día que se efectuara la medición.
RPL	Duración en minutos del tiempo de medición.
RPN	Número de medición en el día.
RPER	Numero de la medición desde que fue definida.
DATE	Fecha de inicio de la medición (AAMMDD).
TIME	Hora de inicio de la medición (HHMM).
PLOAD	Porcentaje de carga del procesador.
OFFDO	Promedio de llamadas originadas/seg. ofrecidas.
OFFDI	Promedio de llamadas entrantes/seg. ofrecidas.
FTCHDO	Promedio de llamadas originadas/seg. procesadas.

FTCHDI Promedio de llamadas entrantes/seg. procesadas

2.1.4 COMANDOS DE MEDICION DE CALIDAD DE SERVICIO.

A.- Comandos e Impresos

Comandos:

SQENE	Estadísticas de calidad de servicio, fin de exclusión de direcciones numéricas.
SQENI	Estadísticas de calidad de servicio, inicio de exclusión de direcciones numéricas.
SQENP	Estadísticas de calidad de servicio, impresión de series numéricas excluidas.
SQMPI	Estadísticas de calidad de servicio, iniciación del programa de medida.
SQMPP	Estadísticas de calidad de servicio, impresión del programa de medida.
SQTLE	Estadísticas de calidad de servicio, finalización de los límites de tiempo.
SQTLI	Estadísticas de calidad de servicio, inicio de los límites de tiempo.
SQTLP	Estadísticas de calidad de servicio, impresión de los límites de tiempo.
SQREP	Estadísticas de calidad de servicio, impresión de los resultados.
TRIDP	Identidades de grabación de datos, imprimir.
TRTSE	Tabla de tiempo de grabación de tráfico, finalizar
TRTSI	Tabla de tiempo de grabación de tráfico, iniciar.
TRTSP	Tabla de tiempo de grabación de datos, imprimir.

Impresos:

SERVICE QUALITY STATISTICS MEASURING PROGRAM
 SERVICE QUALITY STATISTICS RESULT
 SERVICE QUALITY STATISTICS TIME LIMITS
 SERVICE QUALITY STATISTICS EXCLUSION OF NRD
 TRAFFIC RECORDING TIME SCHEDULE
 TRAFFIC RECORDING IDENTITY

B.- Ejecución

a. - Solicitar un programa de medida libre

```
<TRIDP  

  TRAFFIC RECORDING IDENTITY  

  MP  

  11  

  END
```

b. - Definir los límites de tiempo.

```
<SQTLI BANSWER=5&10  

  <SQTLI CTIME=30&120
```

c. - Imprimir los límites de tiempo.

```
<SQTLF  

  SERVICE QUALITY STATISTICS TIME LIMITS  

  PDD          BANSWER          CTIME  

  5             5                 30  

  15            15                120
```

d. - Hacer la exclusión de las series que no deben entrar en la medición.

```
<SQENI NRL=2,NRD=57&61&25
```

e.- Imprimir las series numéricas excluidas:

```
<SQENP
SERVICE QUALITY STATISTICS EXCLUSION OF NRD
NRD
57 61 25
END
```

f.- Definir el programa de medición para estadísticas de calidad de servicio.

```
<SQMPI MP=11,QTA=1000,R2=MS50
```

g.- Imprimir los datos de los programas de medición pertenecientes a estadísticas de calidad de servicio.

```
<SQMPP MP=ALL
SERVICE QUALITY STATISTICS MEASURING PROGRAMS
MP      STATE  ID      CTYPR  R1      ADDRESS  R2      QTA
11      INACT  AT-0                      MS50  1000
NMP=4
END
```

h.- Definir el programa de tiempo para el programa de medición.

```
<TRTSP MP=11,TIME=0930,RPL=120,DATE=950702
```

i.- Imprimir los valores del programa de tiempo.

```
<TRTSP MP=11
TRAFFIC RECORDING TIME SCHEDULE
MP      HRP      RPL      DATE      NDAYS  DCAT      TIME
11      1        120      950702    1      ALL      0930
END
```


C.- Interpretación de resultados.**RESULTADO:****SERVICE QUALITY STATISTICS RESULT**

```

MP      STATE      QTA      OVERLAP  SRS
H       WAHUNG      1000
CTYPE  ADDRESS    R1      REGPOS   SAMPLES  FULFIL  QDATE    QTIME
                MS90    46962    5130     1000    950701   1120
INCIAL  PRERE  DDLE  DNANS  BANS  DBUSV  UNOBNR  RCONG  CCONG
1       215  692  8      100   230    11      10     22

DEFAULT  PDD1  PDD2  PDD3  PDD4  BANSW1  BANSW2  CTIME1  CTIME2
0
PDD      BANSWER      CTIME
        5          30
        15         120

END

```

PARAMETROS:

- MP** Identificación del programa de medición.
- STATE** Estado del programa de medición.
- QTA** Cupo máximo de llamadas en la medición.
- OVERLAP** Indica si hubo superposición de tiempos.
- SRS** Indica si durante hubo re arranque del sistema con recarga.
- CTYPE** Tipo de tráfico a medir.
- R1** Ruta entrante.
- ADDRESS** Dirección numérica.
- R2** Ruta saliente.
- REGPOS** Número de llamadas registradas.
- SAMPLES** Número de llamadas controladas por el programa de medición.
- FULFIL** Número de llamadas que cumplen con los sistemas de selección.

QTADATE	Fecha de la última llamada registrada (solo si se completo QTA).
QTATIME	Hora de la última llamada registrada (solo si se completo QTA).
INCDIAL	Número de llamadas con marcación incompleta.
PRERE	Número de llamadas en las que ha ocurrido reposición de B.
BIDLE	Número de llamadas hacia números B libres.
BNANS	Número de llamadas hacia números B que no contestaron.
BANS	Número de llamadas hacia números B que contestaron.
BBUSY	Número de llamadas hacia números B ocupados.
UNOBNR	Número de llamadas hacia números B inobtenibles.
ICONG	Número de llamadas que encontraron congestión en la central.
OCONG	Número de llamadas con congestión fuera de la central.
TEFAULT	Número de llamadas que encuentran falla técnica en la conexión.
PDD	Demora post-marcación (tiempo desde la última cifra marcada hasta el fin de la selección).
BANSWER	Tiempo de contestación de B (tiempo que transcurre desde el fin de la selección hasta la contestación de B).
CTIME	Tiempo de conversación (tiempo que transcurre desde la contestación de B hasta la desconexión).

CAPITULO 3

DISEÑO E IMPLEMENTACIÓN DEL ADMINISTRADOR

3.1 BENEFICIOS DEL ADMINISTRADOR

Los beneficios del Administrador son claros y concisos. Específicamente son el mejoramiento de un sistema de elaboración de informes de Calidad de Servicio, Tráfico de Rutas y de Carga del Procesador.

El sistema anterior ha sido realizado de una manera semiautomática, es decir, los datos eran "bajados" de la red, se creaba un archivo y luego por medio de un sistema operativo llamado EXCEL se realizaban los gráficos estadísticos utilizados en el informe. Por medio de este método los datos pueden ser manipulados a conveniencia del operador.

Por tal motivo un informe realizado por este método no es 100% confiable.

Diseñar un programa que realice el informe de una manera automática ha sido una tarea extremadamente difícil, debido a los diferentes tipos de formatos de cada uno de los informes. Se utilizó los programas VISUAL BASIC 3.0, Lenguaje C, y ACCESS 2.0 siendo los beneficios del Administrador los siguientes:

- Medición de calidad de servicio
- Medición de carga del procesador
- Medición de tráfico
- Graficación automática de los informes

- Presentación de datos con los parámetros necesarios para la realización del informe
- Opción de impresión.
- Elección de tipo de gráfico (barras, pastel, lineal, etc.)
- Cálculos automáticos de circuitos ideales
- Cálculos automáticos de Erlangs.
- Posibilidad de aumentar rutas o centrales en la base de datos.
- Posibilidad de cambio de grupo de central (central analógica a digital, rural a regional, etc.)

3.2 ESTRUCTURA DE FUNCIONAMIENTO

El funcionamiento del Administrador de acuerdo al esquema diseñado se lo ha diagramado por niveles, de los cuales el Nivel 0 expone, de manera general, los diferentes módulos (Tráfico, Carga y Calidad); el Nivel 1 explica las funciones de cada uno de los módulos sean estos visualización y graficación para Calidad y Tráfico en tanto que para Carga se realiza el proceso completo el cual es cargar, depurar, tabular, realizar cálculos y generar gráficos; el Nivel 2 detalla el proceso que realiza para depurar, calcular , tabular, generar bases, mostrar datos y generar gráficos, siendo estos procesos unicamente en los módulos de Calidad y Tráfico.

3.2.1 DIAGRAMA DE NIVEL 0

ESTRUCTURA DE FUNCIONAMIENTO

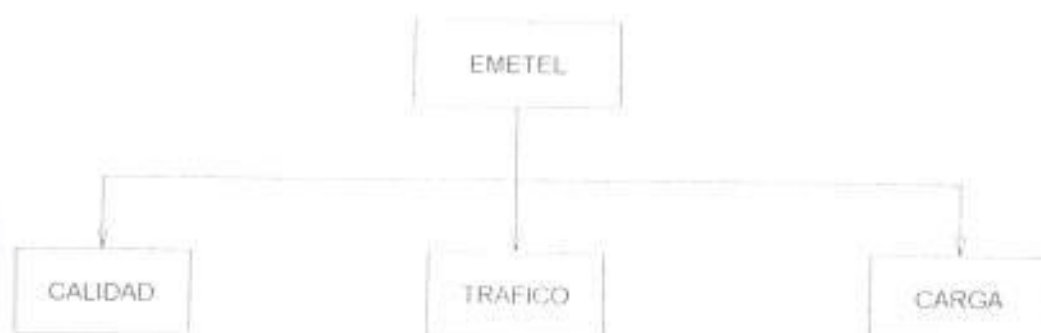


Fig. 3.1

Este es el proceso básico de la aplicación que está formada por tres grandes grupos o módulos los cuales los detallaremos en seguida como son:

- Calidad .- Calidad de servicios telefónico.
- Trafico .- Trafico de rutas telefónicas.
- Carga .- Carga del procesador.

3.2.2 DIAGRAMA DE NIVEL 1 PARA CALIDAD



Fig.32

El proceso de Calidad se realizan los módulos de carga de datos dando opción a ver y graficar los mismos.

3.2.3 DIAGRAMA DE NIVEL 1 PARA TRAFICO



Fig. 3.3

El proceso de Tráfico es en concepto el mismo que el de Calidad con ciertas diferencias que se mostraran más adelante.

3.2.4 DIAGRAMA DE NIVEL 1 PARA CARGA

ESTRUCTURA DE FUNCIONAMIENTO CARGAR



Fig.3.4

La depuración de los archivos se lo realiza en "lenguaje C" la cual trata acerca de la eliminación de aquellas líneas, cantidades que no se utilizan para dejar solamente las cantidades que son necesarias para luego procesarlas y manipularlas, para realizar los gráficos y presentaciones de tablas que se necesitan.

3.2.5 DIAGRAMA DE NIVEL 2 PARA CALIDAD

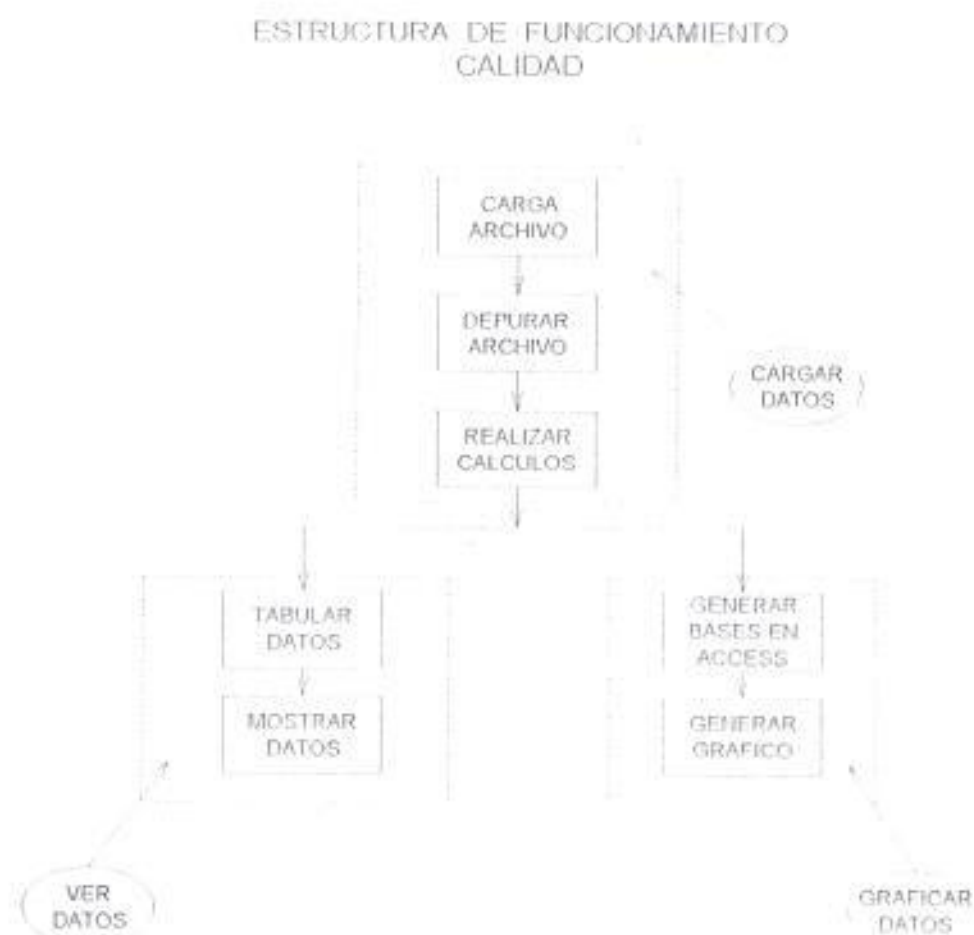


Fig. 3.5

Debo recalcar que los procesos que hemos mencionado como son los de depurar y realizar los cálculos se los realiza en "Lenguaje C" inclusive la misma captura de los archivos de la ERICSSON, se utilizo esta herramienta ya que es la más idónea para este tipo de trabajo por el alcance que tiene por ser un lenguaje a bajo nivel.

3.2.6 DIAGRAMA DE NIVEL 2 PARA TRAFICO

ESTRUCTURA DE FUNCIONAMIENTO TRAFICO

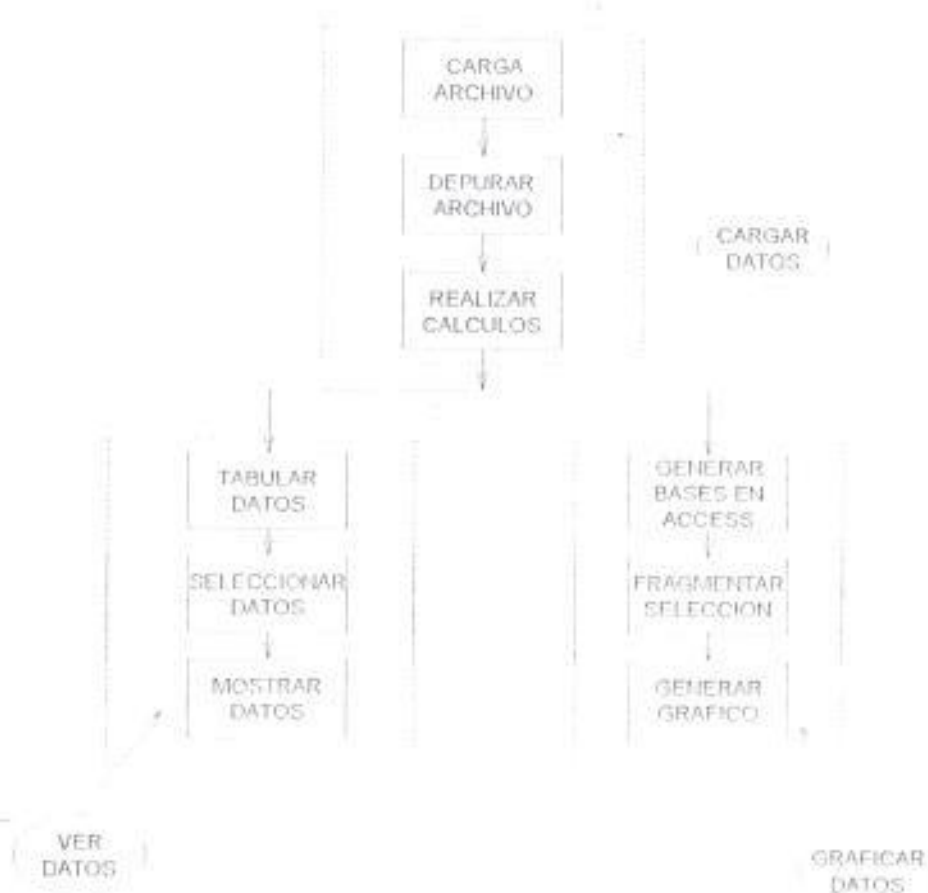


Fig. 3.6

El FromEnd es decir la interface con el usuario se la ha realizado en Visual Basic por la facilidad de manejo de ventanas y menús que esta presenta y el control que posee al utilizar DLL's y junto con OLE 2.0 poder manejar otras aplicaciones como lo es ACCESS nuestro manejador de Bases de Datos y Graficador, utilizada por la facilidad con que estos se manejan.

3.3 CODIGO FUENTE

3.3.1 DECLARACION DE VARIABLES GLOBALES

```

Option Explicit
Global Const DirBin = "c:\emtel\bin"
Global Const DirDatos = "c:\emtel\Datos"
Global Const DirDatos = "c:\Datos"
Global Const DirTmp = "c:\emtel\tmp"
Global Const NMBASE = "Emtel.mdb"
Global Const Repaldo = "c:\pslc\repaldo"
Global bTráfico As Integer
Global bdOpcion As Integer ' 0 es Calidad, 1 es tráfico
Global bdCalidad As Integer ' 0 es Calidad, 1 es tráfico
Global mibase As Database
Global bd_conectarBase As String ' N- NO conectada -S- Si conectada
Global PROBABILIDAD1 As String
Declare Function GetModuleUsage% Lib "Kernel" (ByVal hModulo%)

```

3.3.2 LIMPIAR TEXTO

```

Sub limpia_texto (ugecontrol As Control)
    ugecontrol.Text = ""
End Sub

```

3.3.3 TRANSFERENCIA DE TEXTO

```

Sub Trasl_Texto (Base ()
    Dim X, Z
    Dim str_comando As String
    On Error GoTo err_Trasl_Texto_Base

    If bdOpcion = 0 Then

```

```

    str_comando = DirBin + "MSARUN200.EXE" + DirDatos + "Calidad.mdb"
Else
    str_comando = DirBin + "MSARUN200.EXE" + DirDatos + "NMBASE"
End If
X = Shell(str_comando, 0)
While GetModuleHandle(X) < 0
    Z = DoEvents()
Wend
Exit Sub

err_TrasF_Texto_Base
Resume Next

End Sub

```

3.3.4 VALIDA TEXTO

```

Sub valida_teclas(KeyAscii As Integer, option As Integer)
Select Case option
    Case 0 'Valida Numeros
        If Not ((KeyAscii <= 48 And KeyAscii >= 57) Or KeyAscii = 8 Or KeyAscii = 250 Or KeyAscii =
46) Then
            KeyAscii = 0
        Else
            KeyAscii = Asc("Case 0: In %KeyAscii")
        End If
    Case 1 'Valida letras
        If Not ((KeyAscii <= 65 And KeyAscii >= 90) Or (KeyAscii <= 97 And KeyAscii <= 122) Or
KeyAscii = 46 Or KeyAscii = 193) Or KeyAscii = 201 Or KeyAscii = 203 Or KeyAscii = 209 Or
KeyAscii = 211 Or KeyAscii = 218 Or KeyAscii = 220 Or KeyAscii = 225 Or KeyAscii = 233 Or
KeyAscii = 237 Or KeyAscii = 241 Or KeyAscii = 243 Or KeyAscii = 250 Or KeyAscii = 252 Or
KeyAscii = 8 Or KeyAscii = 32) Then
            KeyAscii = 0
        Else

```



```

    KeyAscii = AscUCase$(Chr$(KeyAscii))
End If
Case 3 'Numeros y Letras
    If Not ((KeyAscii = 48 And KeyAscii <= 57) Or (KeyAscii = 65 And KeyAscii <= 90) Or
    (KeyAscii = 97 And KeyAscii <= 122) Or KeyAscii = 46 Or KeyAscii = 43 Or KeyAscii = 204 Or
    KeyAscii = 205 Or KeyAscii = 209 Or KeyAscii = 211 Or KeyAscii = 218 Or KeyAscii = 220 Or
    KeyAscii = 225 Or KeyAscii = 233 Or KeyAscii = 237 Or KeyAscii = 241 Or KeyAscii = 243 Or
    KeyAscii = 250 Or KeyAscii = 252 Or KeyAscii = 8 Or KeyAscii = 32) Then
        KeyAscii = 0
    Else
        KeyAscii = AscUCase$(Chr$(KeyAscii))
    End If
End Select
End Sub

```

3.3.5 LEE UN PARAMETRO

```

Function Read_Ini (TX_Buscar As String, Tx_Archivo As String, Tx_seccion As String) As String
Dim tx_informa As String
Dim bd_iden As Integer
Dim bd_Existe As Integer

    Read_Ini = ""
    bd_Existe = False
    If Dir(Tx_Archivo, 0) = "" Then bd_Existe = True
    If Not bd_Existe Then Exit Function
    [SECCION]
    tx_informa = String$(55, 0)
    * bd_iden = GetPrivateProfileString(Tx_seccion, TX_Buscar, "", tx_informa, Len(tx_informa),
    Tx_Archivo)
    Read_Ini = Trim$(tx_informa)

End Function

```

3.3.6 REEMPLAZA STRING

```

Function Reemplaza_Str (ByVal tx_string As String, tx_Busca As String, tx_recomp As String) As
String
Dim qx_fin As Integer ' Posicion Final
Dim qx_ini As Integer ' Posicion Inicial
qx_ini = 1
qx_fin = InStr(qx_ini, tx_string, tx_Busca, 0)
While qx_fin
tx_string = Mid$(tx_string, 1, qx_fin - 1) & tx_recomp & Mid$(tx_string, qx_fin + Len(tx_Busca),
Len(tx_string) - qx_ini)
qx_ini = qx_fin + 1
qx_fin = InStr(qx_ini, tx_string, tx_Busca)
Wend
Reemplaza_Str = tx_string
End Function

```

3.3.7 CALCULA FACTORIAL

Es la sub-rutina creada para el calculo factorial, necesitado luego para la obtención de circuitos.

```

Function Fact (x As Integer) As Integer
Dim result As Integer, i As Integer
If x = 0 Then
result = 1
Else
result = 1
For i = 1 To x
result = result * i
Next i
End If
Fact = result
End Function

```

3.3.8 CALCULA NUMERO DE CIRCUITOS

Esta sub-rutina ha sido creada para el calculo de los circuitos ideales los cuales son dados en Erlangs y nosotros los necesitamos en circuitos.

```
Function Calcula_N (E As Integer, A As Integer) As Integer
  Dim result As Integer, i As Integer
  Dim intento As Double, falla As Double
  falla = .001
  For i = 1 To 2000
    intento = (Fact(i) / Fact(i - 1)) * suma(A, i)
    If (intento + falla) = 1 / E Or (intento - falla) = 1 / E Then
      result = i
      Exit For
    End If
  Next i
  Calcula_N = result
End Function
```

3.3.9 PROCESO DE SUMATORIA

Realiza la sumatorio de los términos de la fórmula de obtención de circuitos, la cual se encuentra en los anexos.

```
Function suma (x As Integer, y As Integer) As Double
  Dim i As Integer
  Dim result As Double
  result = 0
  For i = 0 To y
    result = result + (1 / (x + i))
  Next i
  suma = result
End Function
```

3.3.10 PROCESAR CALIDAD

Ordena los parámetros ha utilizarse on los diferentes informes.

```

Sub Procesar_calidad (Path As String, flag As Integer)
    Dim x As Integer, y As Integer
    Dim temp As String, temp1 As String, temp2 As String
    Dim linea As String
    temp = App.Path
    If Right$(temp, 1) = "\" Then temp = temp & "*"
    temp1 = temp & "calidad.exe"
    x = Shell(temp1 & "*" & Path & "*" & temp & "temp.txt", vb)
    Screen.MousePointer = 11
    Do While GetModuleUsage(x) = 0
        y = DoEvents()
    Loop
    Screen.MousePointer = 0
    temp2 = temp & "temp.txt"
    Screen.MousePointer = 0
    On Error GoTo xl:
    Open temp2 For Input Shared As #1
    Line Input #1, linea
    Line Input #1, linea
    FRM_Principal.Text1.Text = "TU CHA " & linea & " CRLF"
    FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & "R      FUJH.  PRUF
BNANS  BANS  BBUSY  TTFACUL" & " CRLF"
    Do While Not EOF(1) And flag_cancelar = False
        Line Input #1, linea
        Line Input #1, linea
        FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & Formato_linea(linea)
        Line Input #1, linea
        Line Input #1, linea
        FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & Formato_linea(linea)
        Line Input #1, linea
        Line Input #1, linea
    
```

```

FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato + Linea(linea)
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato + Linea(linea)
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato + Linea(linea)
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato + Linea(linea)
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato + Linea(linea)
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + CR LF
y = DoEvents()
Loop
Close
Screen.MousePointer = 0
Flag_archivo = True
Flag_grabar = True
FRM_Principal.Text1.Enabled = True
Exit Sub
xl:
MsgBox "No se pudo leer el archivo", vb, "Atencion!"
Close
Screen.MousePointer = 0
Exit Sub
End Sub

```

3.3.11 PROCESAR CARGA

Realiza los calculos de los parametros utilizados en la carga del procesador siendo estos, BHCA Y BHC.

```
Sub Procesar_Carga (Path As String, flag As Integer)
```

```

Dim x As Integer, y As Integer
Dim temp As String, temp1 As String, temp2 As String
Dim linea As String
temp = App.Path
If Right$(temp, 1) = "\" Then temp = temp & "*"
temp1 = temp & "carga.exe"
x = Shell(temp1 & "*" & Path) & "*" & temp & "temp.txt", 6)
Screen.MousePointer = 11
Do While GetModuleUsage(x) = 0
    y = DoEvents()
Loop
Screen.MousePointer = 0
temp2 = temp & "temp.txt"
Screen.MousePointer = 0
On Error Go To x2
Open temp2 For Input Shared As #1
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = "FECHA: " & linea & CRLF
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & "HMI     PLVAD     OFFDO"
OFFDI     FEHDI     FEHDI" & CRLF
Do While Not EOF(1) And flag = com.dat = False
    Line Input #1, linea
    Line Input #1, linea
    FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & Format$(linea, linea)
    Line Input #1, linea
    Line Input #1, linea
    FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & Format$(linea, linea)
    Line Input #1, linea
    Line Input #1, linea
    FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & Format$(linea, linea)
    Line Input #1, linea
    Line Input #1, linea
    FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & Format$(linea, linea)

```



```

Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato.Linea(linea)
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato.Linea(linea)
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato.Linea(linea)
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + CRLF
y = DoEvents()
Loop
Close
Screen.MousePointer = 0
Flag_archivo = True
Flag_grabar = True
FRM_Principal.Text1.Enabled = True
Exit Sub
x2:
MsgBox "No se puede leer el archivo", 16, "Atencion!"
Close
Screen.MousePointer = 0
Exit Sub
End Sub

```

3.3.12 PROCESAR PICOS

El proceso es analizar y escoger el valor mas alto de la columna de TRAFF comparadas cada ruta hora por hora de cada TRO (0-3).

```

Sub Procesar_Pico (Path As String, flag As Integer)
Dim x As Integer, y As Integer
Dim temp As String, temp1 As String, temp2 As String
Dim linea As String
temp = App.Path

```

```

If Right$(temp, 1) = "*" Then temp = temp + "*"
temp1 = temp + ".log.exe"
x = Shell(temp1 + " " + Path + " " + temp + ".temp.txt" + Str$(FRG), 6)
Screen.MousePointer = 11
Do While GetModuleUsage(x) = 0
    y = DoEvents()
Loop
Screen.MousePointer = 0
temp1 = temp + ".pic.exe"
x = Shell(temp1 + " " + temp + ".temp.txt" + " " + temp + ".temp1.txt", 6)
Screen.MousePointer = 11
Do While GetModuleUsage(x) = 0
    y = DoEvents()
Loop
Screen.MousePointer = 0
temp1 = temp + ".indico.exe"
x = Shell(temp1 + " " + temp + ".temp1.txt" + temp + ".temp2.txt" + Str$(Probabilidad), 6)
Screen.MousePointer = 11
Do While GetModuleUsage(x) = 0
    y = DoEvents()
Loop
Screen.MousePointer = 0

temp2 = temp + ".temp2.txt"
Screen.MousePointer = 0
On Error GoTo x1
Open temp2 For Input Shared As #1
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = "TECNA " + linea + CRLF
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + "R          EXISTENTE IDEAL
OFRECHO" + CRLF
Do While Not EOF(1) And flag_cancelar = False

```

```

Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato_linea(linea)
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato_linea(linea)
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + Formato_linea(linea)
Line Input #1, linea
Line Input #1, linea
FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + CPLE
y = DoEvents()
Loop
Close
Screen.MousePointer = 0
Flag_archivo = True
Flag_grabar = True
FRM_Principal.Text1.Enabled = True
Exit Sub
s4
MsgBox "No se pudo leer el archivo", 16, "Atencion!"
Close
Screen.MousePointer = 0
Exit Sub
End Sub

```

3.3.13 PROCESAR RUTAS

Esta sub-rutina se va a guardar a un archivo de iguales características que el archivo original de una hora con la variante de que cada ruta tendrá una columna adicional en la cual constara de que hora fue escogida (hora pico).

```
Sub Procesar_Rutas (Path As String, flag As Integer)
```

```

Dim x As Integer, y As Integer
Dim temp As String, temp1 As String, temp2 As String
Dim linea As String
temp = App.Path
If Right$(temp, 1) <> "." Then temp = temp & "."
temp1 = temp & "trafico.exe"
x = Shell(temp1 & "*" & Path & "*" & temp & "temp.txt" & Str$(Probabilidad), 6)
Screen.MousePointer = 11
Do While GetModuleUsage(x) > 0
    y = DoEvents()
Loop
Screen.MousePointer = 0
temp2 = temp & "temp.txt"
Screen.MousePointer = 0
On Error GoTo x3
Open temp2 For Input Shared As #1
    Line Input #1, linea
    Line Input #1, linea
    FRM_Principal.Text1.Text = "FECHA: " & linea & vbCrLf
    FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & "R          EXISTENTE          IDEAL
OFRECIDO" & vbCrLf
    Do While Not EOF(1) And flag = circular = False
        Line Input #1, linea
        Line Input #1, linea
        FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & Formato(linea,linea)
        Line Input #1, linea
        Line Input #1, linea
        FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & Formato(linea,linea)
        Line Input #1, linea
        Line Input #1, linea
        FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & Formato(linea,linea)
        Line Input #1, linea
        Line Input #1, linea
        FRM_Principal.Text1.Text = FRM_Principal.Text1.Text & Formato(linea,linea)

```

```

    FRM_Principal.Text1.Text = FRM_Principal.Text1.Text + CR LF
    y = DoEvents()
Loop
Close
Screen.MousePointer = 0
Flag_archivo = True
Flag_grabar = True
FRM_Principal.Text1.Enabled = True
Exit Sub
x3
MsgBox "No se puede leer el archivo", tv, "Atencion!"
Close
Screen.MousePointer = 0
Exit Sub
End Sub

```

3.3.14 CONECTAR BASE

Es la conexión de la base con el programa, subrutina en lenguaje C.

```

Function conectar_Base (NMBASE1 As String) As Integer
    On Error GoTo err_connectBase
    bd_connectarBase = "N" ' No esta conectada
    conectar_Base = False
    'Set mywork = WildSpecs(0)
    Set mdbase = OpenDatabase(DirDatos + NMBASE1)
    bd_connectarBase = "S" ' Si esta conectada
    conectar_Base = True
    Exit Function
err_connectBase:
    MsgBox CStr(Err) + " " + Err
    MsgBox "Error a conectarse a la base de Datos"
    Exit Function
End Function

```

3.3.15 RECIBIR DATOS

Es la recepción de los datos del programa.

```
Sub Recibir_Datos (x As Form)
    Dim i As Integer
    For i = 1 To FRM1.Graph1.NumPoints
        x.Combo1.AddItem Str$(i)
    Next i
    For i = 1 To FRM1.Graph1.NumSets
        x.Combo2.AddItem Str$(i)
    Next i
End Sub
```

3.3.16 TABLA TEMPORAL

Es la tabla donde llegan los datos y se almacenan temporalmente para la visualización de los mismos si son requeridos.

```
Function GetItemTabla (criterio As String, item As String) As String
    Dim mitabla As Dynaset
    Dim qs_i As Integer
    Dim qx Ion As Long
    On Error GoTo err_GetItemTabla
    GetItemTabla = ""

    Set mitabla = mibase.CreateDynaset(criterio)
    If mitabla.RecordCount = 0 Then
        GetItemTabla = mitabla.Fields(item)
    End If
    Exit Function
err_GetItemTabla:
    MsgBox "Error al obtener Registro"
End Function
```

CAPITULO 4

OPERACIÓN DEL ADMINISTRADOR Y RESULTADOS

4.1 DESCRIPCIÓN GENERAL

El presente capítulo tiene como finalidad realizar una presentación de las principales pantallas del Administrador de Medición de Tráfico y Calidad de Servicio de Centrales Locales ERICSSON de la red Metropolitana de Guayaquil.

Dicha presentación de las diversas ventanas de nuestro programa, vienen acompañadas de una breve explicación de sus respectivas funciones sirviendo de guía para el operador en la obtención de los resultados que se requiera.



Fig.4.1

La ventana mostrada es la presentación del programa, dando al operador un enfoque inicial del Administrador.

Aquí podemos observar una ventana que nos da la elección de continuar con los requerimientos deseados, es decir el de correr el programa, o la opción de salir del mismo:



Fig. 4.2

4.2 MULTISELECCION PARA MEDIR CALIDAD DE SERVICIO

A continuación tenemos la pantalla de opciones, las cuales son cuatro: Calidad de servicio, Tráfico, Carga del procesador y Salir del programa.

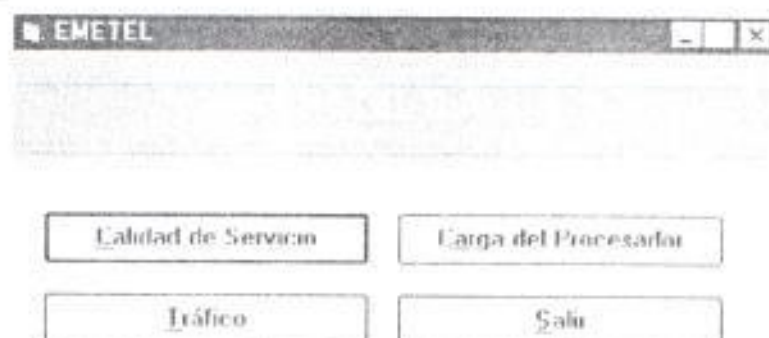


Fig. 4.3

En este caso nuestra elección será la de Calidad de Servicio.

ELECCION DE FUENTE DE DATOS

Poseemos la opción de abrir un archivo existente en cualquier parte, ya sea esta el disco duro del computador o un diskette del operador.



Fig. 44

VENTANA PARA CARGAR DATOS

Habiendo elegido el archivo correspondiente a calidad de servicio procedemos a la elección de: Cargar datos (esta opción nos indica que el programa recoge los datos con el formato enviado por la red e inmediatamente realiza el procedimiento correspondiente a la opción de calidad de servicio y el formato requerido por nosotros es realizado. Luego que el programa ha terminado debemos escoger la opción de presentar datos.) , Presentar datos es aquella que nos muestra los datos requeridos por el operador para su observación y finalmente tenemos la selección de salir, la cual nos expulsa del programa directamente.

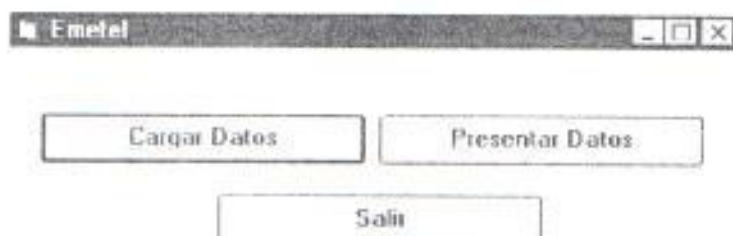


Fig. 4.5

VENTANA DE SELECCIÓN POR CENTRAL O POR RUTA

La ventana a continuación es la que nos muestra elecciones tales como: Los datos que se desea observar serán por central, en general, o si se desea observar los datos de calidad por cada una de las rutas de una central predeterminada por el operador en esta misma ventana.

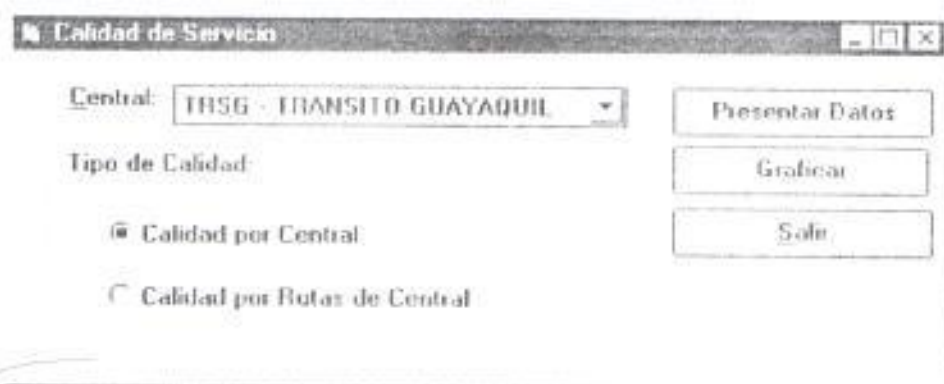


Fig. 4.6

PRESENTACION DE DATOS

En este caso se encuentran los datos de la central Tránsito de Guayaquil, por medio de la selección de Presentar datos. Podemos ver que además existe la posibilidad de salirnos del programa en cualquier momento deseado por el operador.

Como podemos observar los datos de calidad de servicio son mostrados en un nuevo formato creado por el programa siendo estos los principales para la elaboración de los informes requeridos.

Calidad de Servicio
TRSG TRANSITO GUAYAQUIL

Fecha: 17/10/1997

Ruta	Piers	Roms	Ranc	Rmsy	(Total)
BAYO	22.00	1.00	22.00	1.00	0.00
SALTO	13.00	0.00	13.00	1.00	0.00
TIBO	4.50	0.00	22.00	0.00	4.00
BALLO	11.00	0.00	14.00	18.00	1.00
STALLO	6.00	0.00	19.00	16.00	22.00
SAMBRO	16.00	0.00	40.00	1.00	4.00
ETASO	7.00	0.00	12.00	25.00	1.00
PAUSO	14.00	0.00	28.00	1.00	0.00
MOTOSO	25.00	2.00	44.00	28.00	1.00
VIRACO	21.00	2.00	140.00	46.00	1.00
LATINO	2.00	0.00	20.00	17.00	0.00
BARDO	3.00	0.00	11.00	10.00	0.00
OSERO	15.00	0.00	27.00	2.00	1.00
CRIDO	0.00	0.00	17.00	1.00	1.00
MOTCO	5.00	0.00	70.00	12.00	4.00
BALZU	3.00	0.00	26.00	27.00	2.00
ECUDO	11.00	0.00	22.00	21.17	5.00
CATRO	16.00	1.00	25.00	1.00	11.00
DATRO	4.00	0.00	127.00	178.00	47.00
MORDO	4.00	1.00	11.00	1.00	1.00
FALLO	1.00	0.00	23.00	27.00	1.00

Fig. 47

SELECCIÓN DE GRAFICO

Luego de haber analizado la pantalla de datos de Calidad de Servicio seleccionamos la opción de graficar aquellos datos mostrados anteriormente, sean estas por central o por rutas de central.

Calidad de Servicio

Central:

Tipo de Calidad:

Calidad por Central

Calidad por Rutas de Central

Presentar Datos

Graficar

Salir

Fig. 48

GRAFICO POR CENTRAL

Seleccionándola obtendremos la pantalla a continuación, la cual nos muestra el informe en un cuadro estadístico a nuestra elección (generalmente es en forma de pastel 3D). Como podemos observar se encuentran todos los parámetros necesitados por EMETEL para realizar un informe de Calidad de Servicio de sus rutas y centrales Ericcson en Guayaquil.

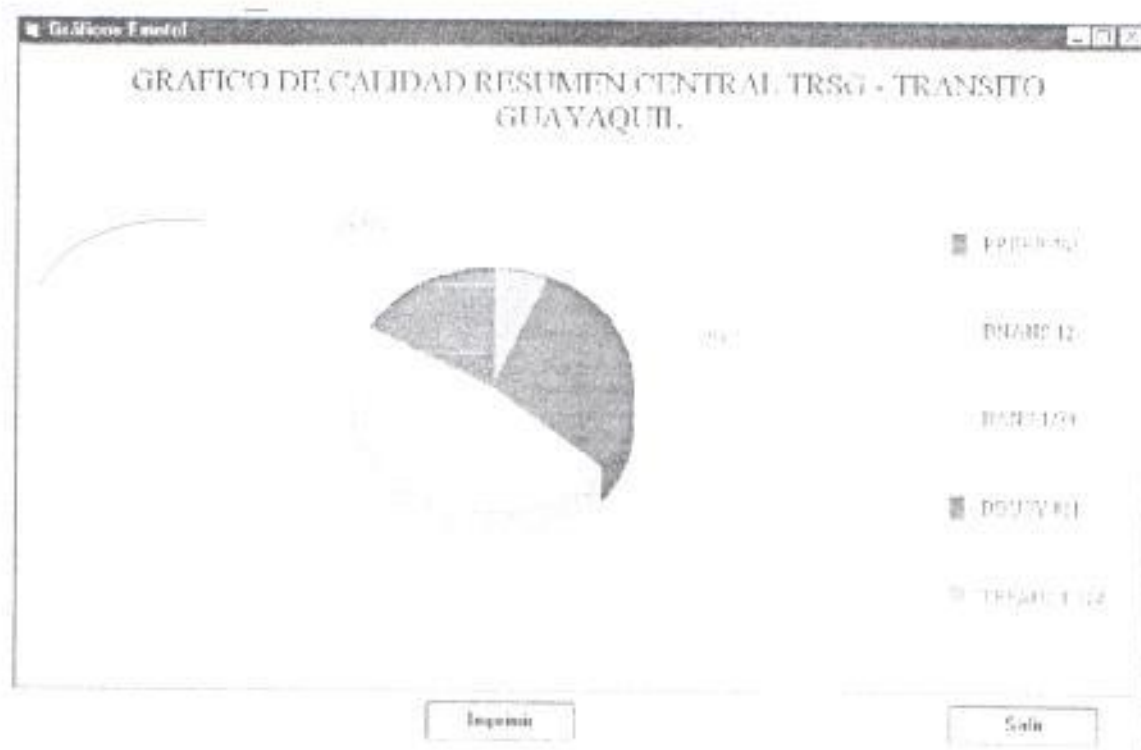


Fig. 49

Como vemos en este gráfico, el operador puede directamente mandar a imprimir la estadística sin necesidad de utilizar otro sistema operativo compatible. La opción de salir del programa esta presente en todo momento para mayor comodidad del usuario.

GRAFICO POR RUTAS

Ahora contamos con la selección de observar gráficamente los mismos parámetros requeridos para el informe de calidad de servicio pero por cada una de las rutas de la central escogida.

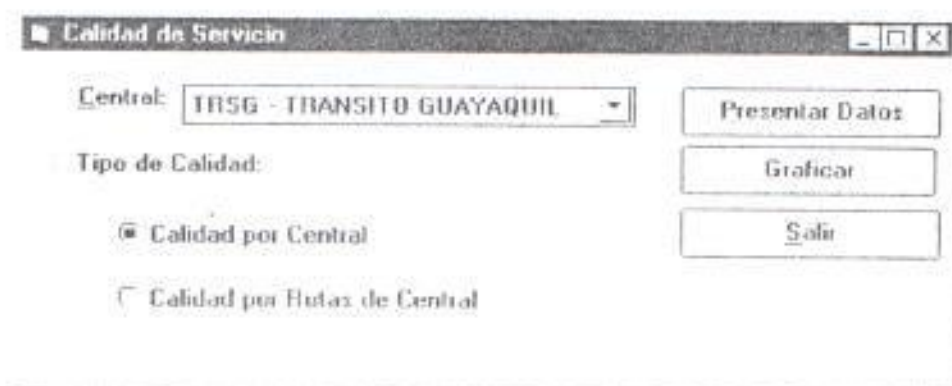


Fig. 4.10

Luego de colocarnos en la selección de Calidad por rutas de central, procedemos a presionar GRAFICAR, para así obtener el gráfico deseado de las rutas de nuestra central escogida.

Tomando en cuenta que el número de rutas de una central puede ser mayor al posible presentado por un solo gráfico existe la opción de SIGUIENTE así como también la de ANTERIOR las cuales muestran un gráfico de las primeras rutas y de las últimas rutas de dicha central.

La opción de IMPRIMIR se encuentra a la disposición del operador.

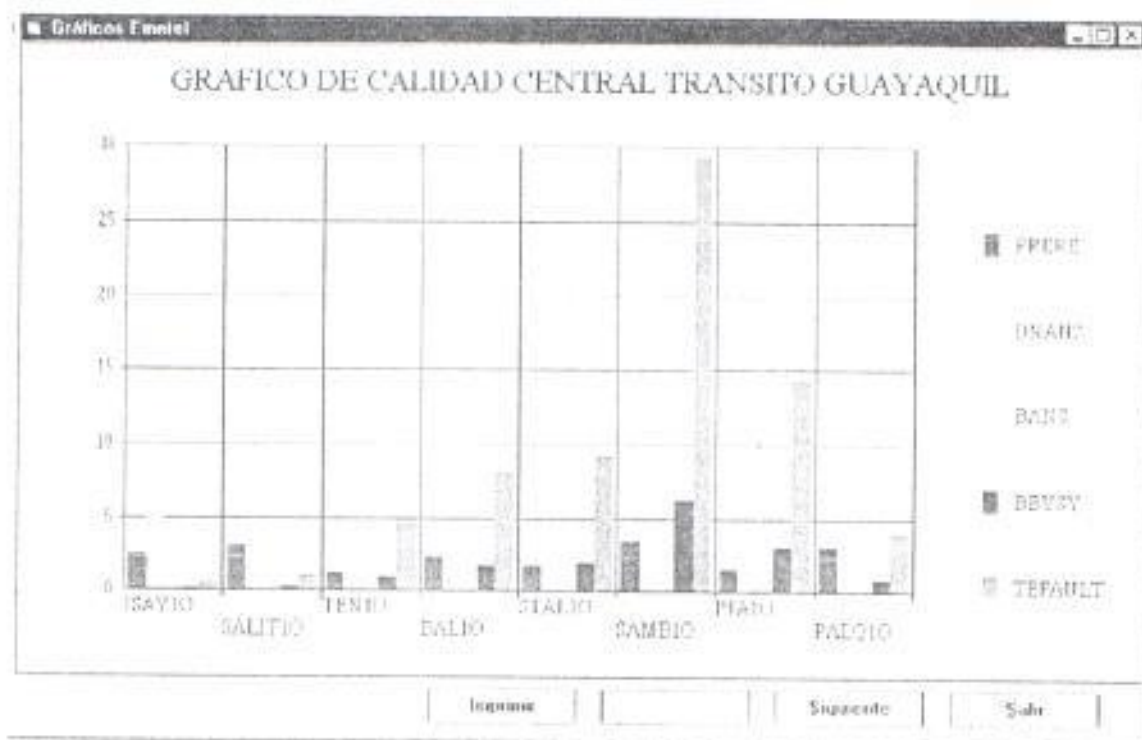


Fig. 4.11

4.3 MULTISELECCION PARA MEDIR TRAFICO

Como podemos observar la ventana de selección múltiple, nos indica a cual selección escoger, en este caso en particular será la de TRAFICO.

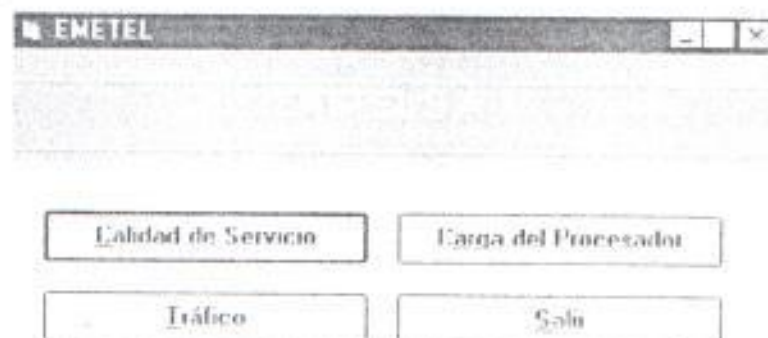


Fig. 4.12

Como en el caso de Calidad de Servicio, tenemos opciones de buscar nuestro archivo de datos con el cual trabajar.

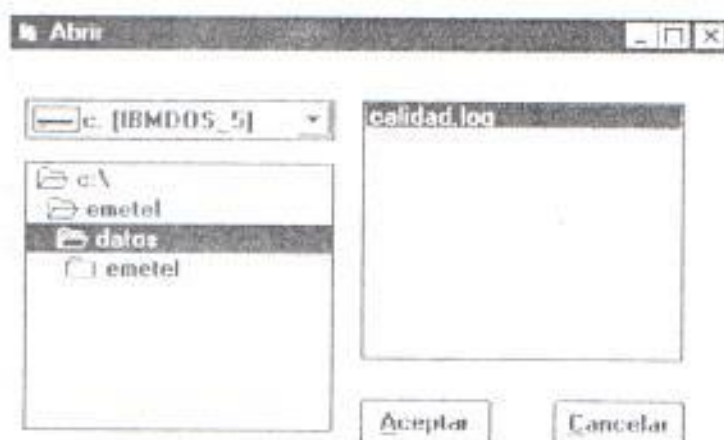


Fig. 4.13

Ahora podemos con la siguiente ventana escoger como queremos que sean mostrados nuestros datos, ya sean esto de una central específica, entrantes o salientes, en hora pico o no, etc.

VENTANA DE SELECCIONES



Fig. 4.14

En esta ventana el operador tendrá la opción de elegir que central requiere para visualizar sus datos, aquí podrá además escoger el grupo (analógicas o digitales; entrantes o salientes; probabilidad de 1% o 2%), y la fecha.

TRAFICO POR HORA PICO

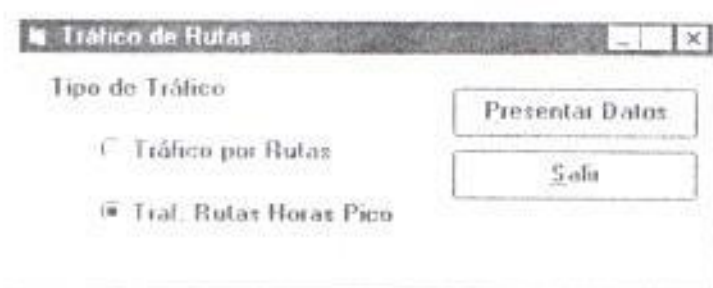


Fig. 4.15

Seleccione tráfico por rutas hora pico si su deseo es obtener los datos en dichas horas.

VENTANA DE HORAS PICO



Fig. 4.16

Como observamos en esta ventana, difiere a la de tráfico por rutas en la elección de la fecha de inicio y de fin.

PRESENTACION DE DATOS

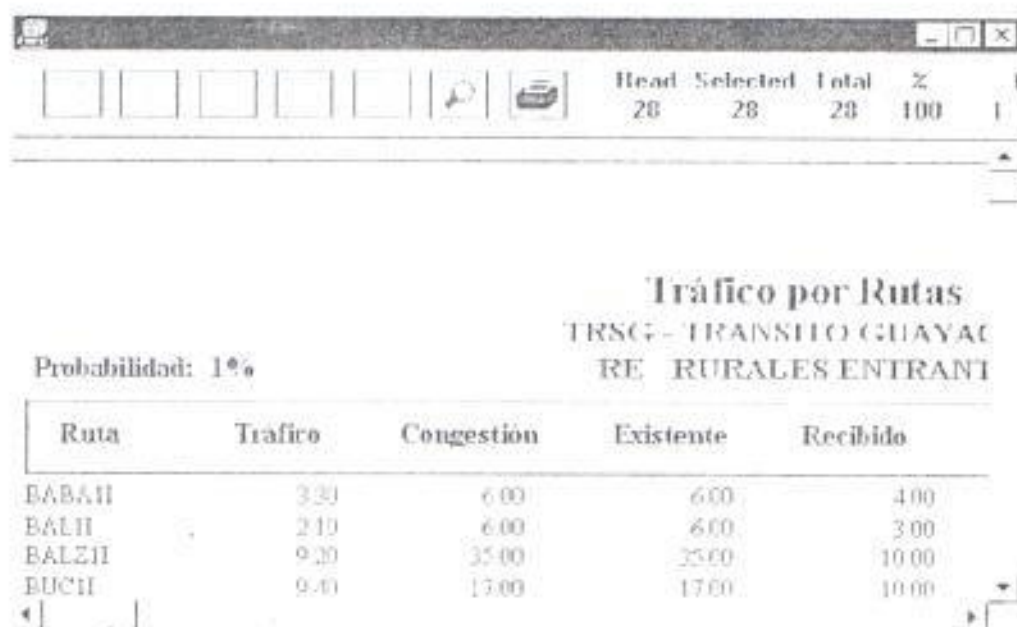


Fig. 4.17

Una vez que se presiona la opción presentación de datos, se visualizará la presente pantalla en la cual se detallan todos los parámetros necesarios que han sido procesados para la elaboración de informes de tráfico. En este caso se muestra las rutas rurales entrantes de la central Tránsito Guayaquil.

Cabe indicar que también se puede visualizar la probabilidad con la cual han sido procesados los datos, en este caso es del 1%.

También se podrá revisar la fecha en la cual han sido obtenidos los datos, por lo que en la esquina superior derecha aparece.

GRAFICO DE RUTAS

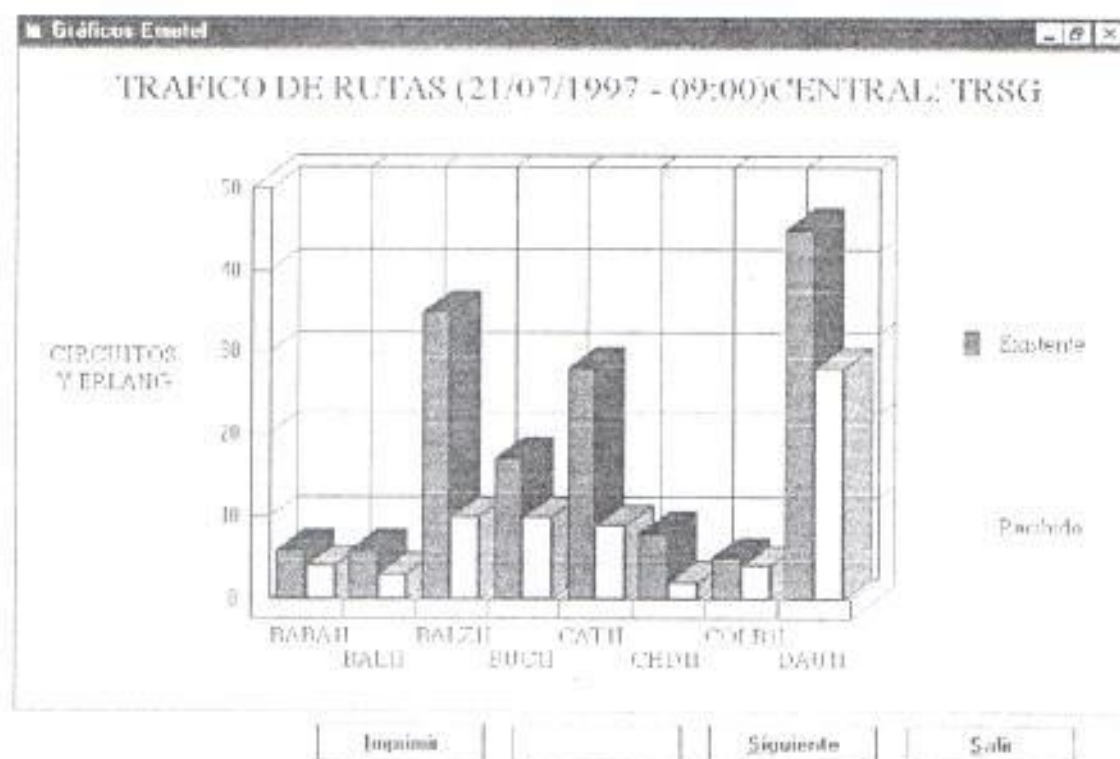


Fig. 4.18

En la presente ventana se puede visualizar el gráfico que se obtiene del procesamiento de datos de la central Tránsito Guayaquil en el grupo de rutas rurales entrantes.

Este grupo de rutas rurales entrantes se podrán presentar en grupos de ocho rutas por pantallas por lo que si existen más de 8 rutas rurales entrantes en esa central, se subdividirá en más de una pantalla.

Cabe indicar que en todos los grupos de rutas pertenecientes a una central cualquiera, solamente se visualizarán dos barras por ruta las cuales son: tráfico en Erlangs y circuitos existentes. La barra de circuitos ideales no se puede presentar porque en las rutas entrantes no se puede tabular los datos de congestión.

PRESENTACION DE OTRAS RUTAS

Head Selected Total %
9 9 9 100 1

Tráfico por Ruta
TRSG - TRANSITO GUA
IE - INTERNACIONAL E

Probabilidad: 1%

Ruta	Trafico	Congestion	Existente	Recibido
ATLNDI	57.50	120.00	120.00	58.00
CHILEI	8.70	22.00	22.00	0.00
INTERI	96.50	247.00	247.00	97.00
INTI	0.00	58.00	58.00	0.00
TCLBIAI	0.00	30.00	30.00	0.00

Fig. 4.19

En este caso se tienen rutas internacionales entrantes de la central Tránsito Guayaquil.

Como en el caso anterior se podrán visualizar todas las características del procesamiento de esos datos.

GRAFICOS DE OTROS TIPOS DE RUTAS

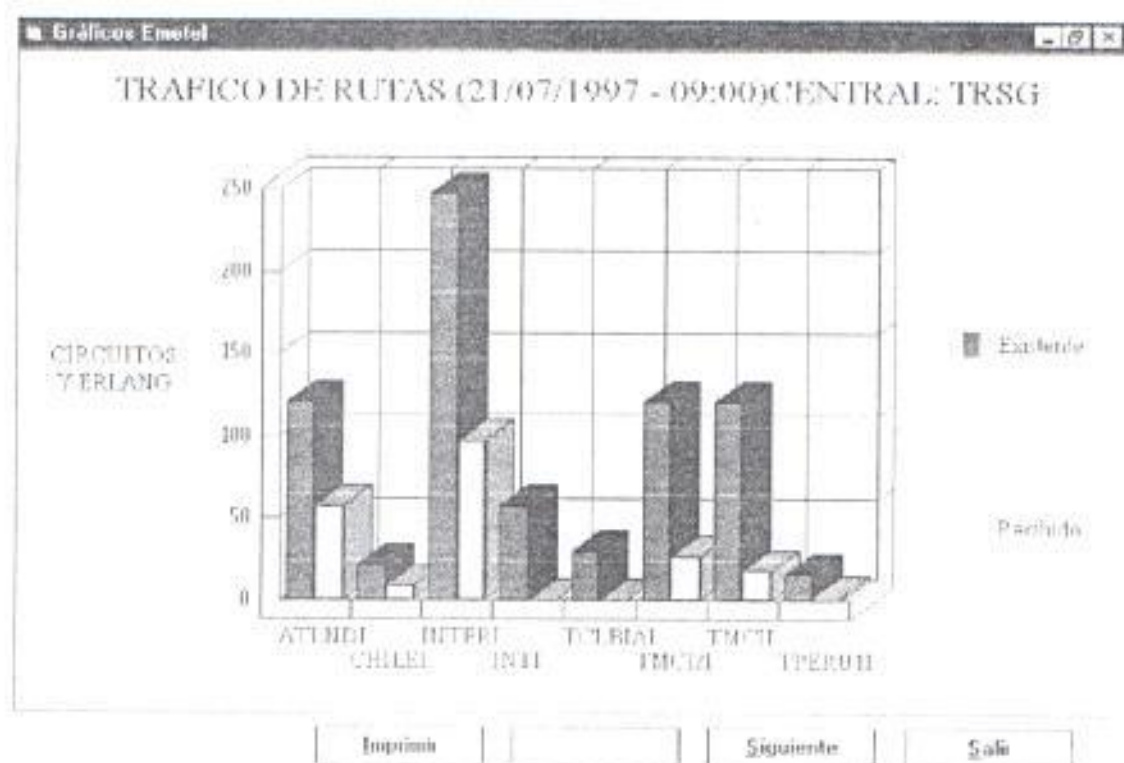


Fig. 4.20

Aquí se visualizan las rutas internacionales entrantes de la Central Tránsito Guayaquil.

PRESENTACION DE MAS RUTAS

Tráfico por Rutas
TRSC - TRANSITO GUAYAQUIL
IS - INTERNACIONAL SALIEI

Probabilidad:

Ruta	Tráfico	Congestión	Existente	Ofrecido
ATLDO	19.90	0.00	120.00	20.00
CHILEO	4.90	0.00	22.00	5.00
ENTERO	44.90	0.00	247.00	45.00
INTO	9.10	0.00	50.00	10.00

Fig. 4.21

En este caso se observan las rutas internacionales salientes de la central Tránsito Guayaquil.

GRAFICO DE MAS RUTAS SALIENTES

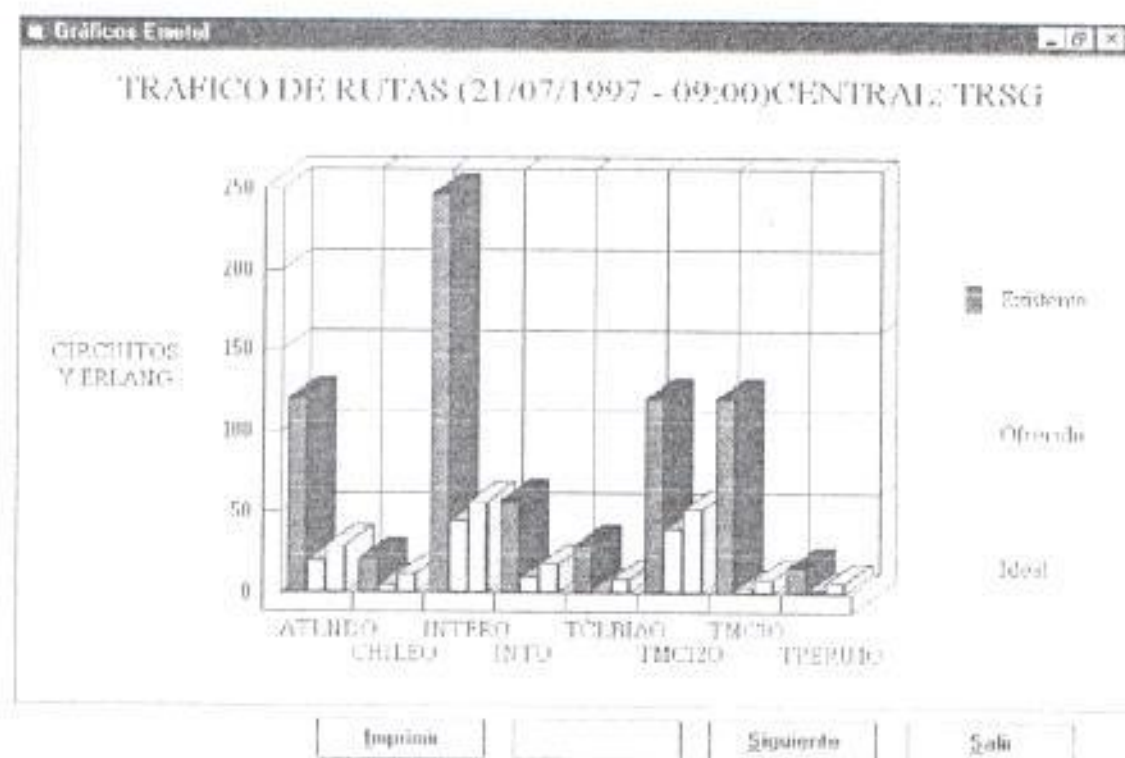


Fig.422

En este caso se observan rutas internacionales salientes de la central Tránsito Guayaquil.

4.4 MULTISELECCION PARA MEDIR CARGA DEL PROCESADOR

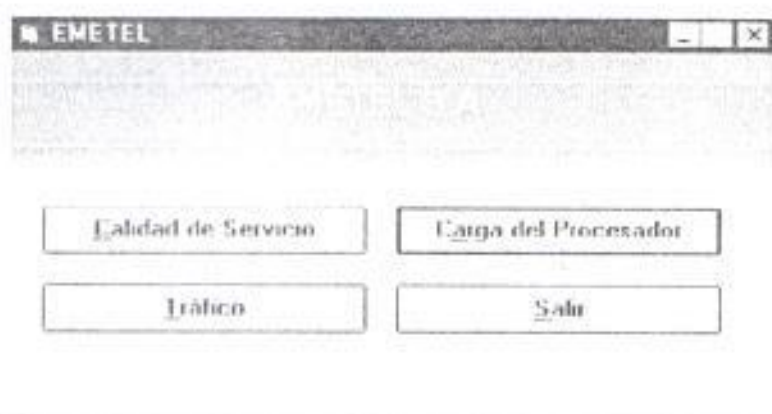


Fig. 4.23

El último tipo de informe que nuestro programa puede procesar es la carga del procesador. Es así que la presente ventana tiene la opción de procesar informes de carga del procesador, por lo que se deberá ingresar en ese icono.

OPCIONES DE EJECUCION

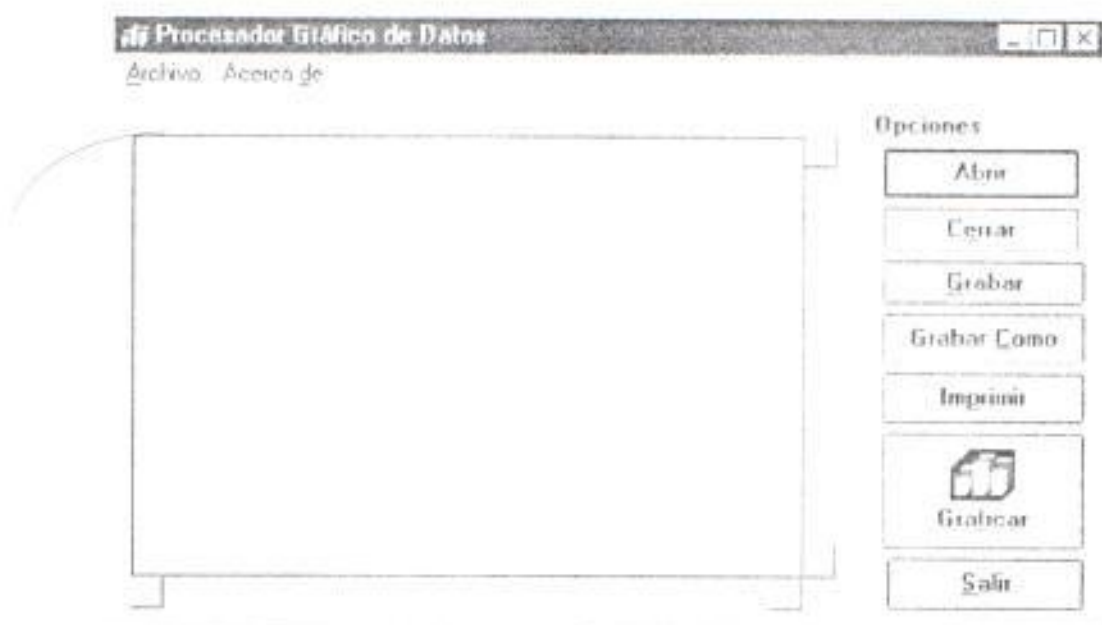


Fig. 4.24

Una vez que se haya seleccionado a un archivo existente como la fuente de datos, entonces se deberá abrir un archivo, presionando la opción ABRIR.

ELECCION DE FUENTE DE DATOS

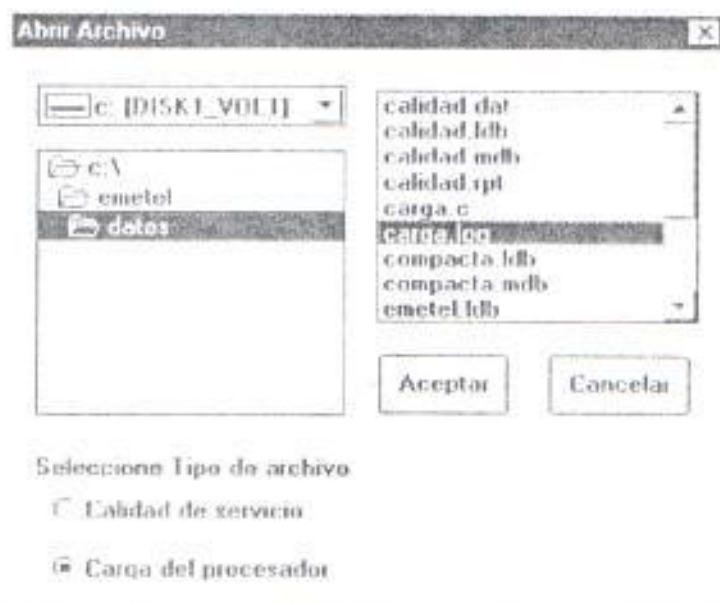


Fig. 4.25

Luego de que se haya seleccionado el archivo correcto con datos de carga del procesador, se procederá a seleccionar el tipo de informe que se desee elaborar, ya sea carga del procesador o calidad de servicio.

PRESENTACION DE DATOS TABULADOS

FECHA	PLUAD	HEFDE	HEFDE	HEFDE
0000	26	0.0	2.8	0.0
0015	25	0.0	2.2	0.0
0030	25	0.0	1.8	0.0
0045	24	0.0	1.4	0.0
0100	24	0.0	1.2	0.0
0115	25	0.0	1.1	0.0
0130	24	0.0	0.9	0.0
0145	24	0.0	1.0	0.0
0200	25	0.0	0.9	0.0
0215	25	0.0	0.9	0.0
0230	28	0.0	0.9	0.0
0245	24	0.0	0.7	0.0
0300	25	0.0	0.7	0.0
0315	24	0.0	0.8	0.0

Opciones

Abrir

Cerrar

Grabar

Grabar Como

Imprimir

Graficar

Salir

Fig. 4.26

Una vez que se han procesado los datos, se muestran en la ventana de presentación. En esta pantalla se pueden visualizar los datos que van a ser graficados en el informe final. En esta pantalla también existen las opciones de grabar, de impresión y de graficar. Es decir que una vez que seleccionemos una de esas opciones se procederá a ejecutar dichas acciones.

SELECCION DE TIPO DE GRAFICO

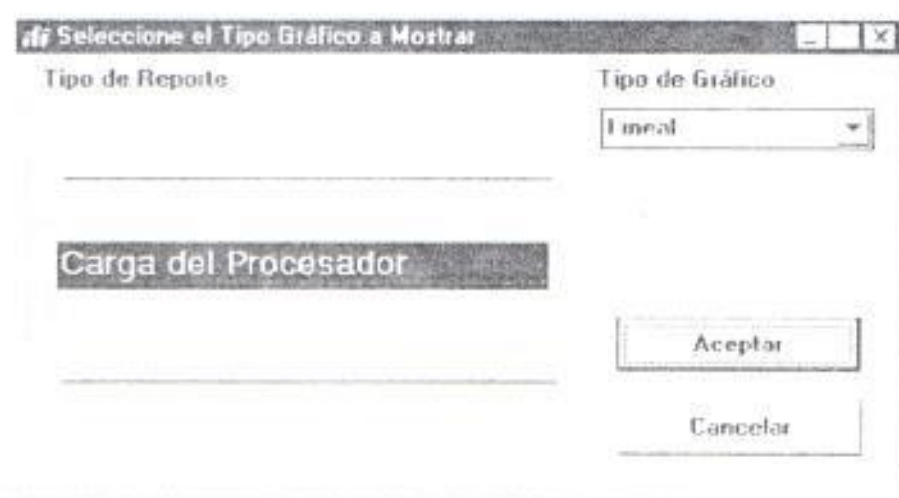


Fig.4.27

Si se seleccionó la opción de graficar entonces se deberá elegir la forma de gráfico que se va a usar. En este caso se deberá escoger el tipo lineal, para poder visualizar correctamente los datos de carga del procesador. Después de haber elegido el tipo de gráfico se debe presionar la tecla aceptar.

GRAFICO PLOAD VS. TIEMPO (CARGA DEL PROCESADOR)



Fig. 428

Cuando se haya apretado la tecla aceptar, aparecerá el gráfico de la carga del procesador del archivo escogido para el informe.

En este gráfico se observan las horas del día vs. La carga que ha tenido el procesador en esas horas. Es por esto que este gráfico nos da una imagen clara de cuáles son las horas más congestionadas del día.

GRAFICO BHCA, BHC VS. TIEMPO (CARGA DE TRAFICO)



Fig. 4.29

En el presente gráfico también se visualizan los datos de carga en diversas horas del día. Cabe indicar que en esta ventana se puede optar por avanzar al siguiente gráfico o retroceder al gráfico anterior. Es así que esta presentación es lo más dinámica posible para el usuario.

4.5 ADMINISTRADOR DE LA BASE DE DATOS

El presente administrador de las bases de datos surge como respuesta a la necesidad de mantener actualizada la cantidad de centrales tipo AXE así como la cantidad de rutas entrantes o salientes que se conecten a cualquier central. También sirve para actualizar los grupos de centrales en alguna subdivisión que con el futuro aparezca.

Con este administrador se asegura que el operador del programa de Procesamiento de Datos para elaboración de informes de tráfico, calidad de servicio y de carga del procesador, no encuentre obstáculos en sus reportes.

4.5.1 VENTANA DE PRESENTACION

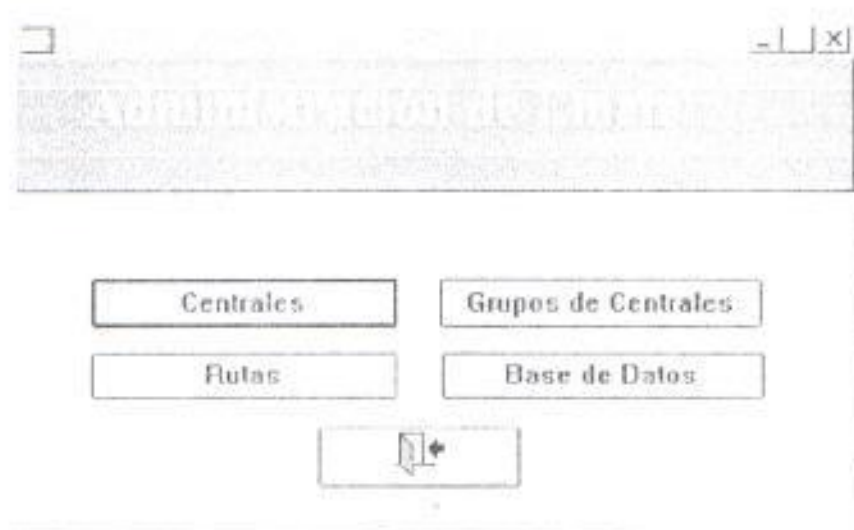


Fig. 430

En la presente ventana de presentación se describen las bases de datos que se pueden administrar tales como: centrales, rutas y grupos de centrales. También aparece la opción Base de Datos que es la que se usa para reparar las bases de datos una vez que se hayan actualizado.

4.5.2 SELECCION DE BASE POR CENTRALES

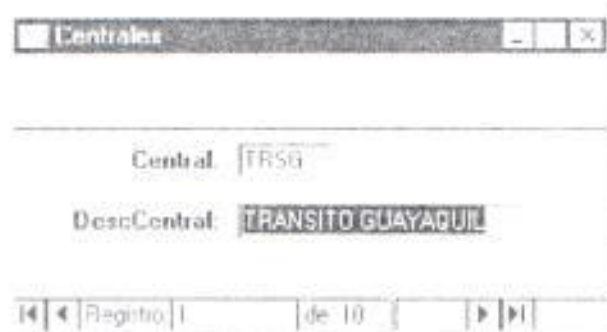


Fig. 431

En la selección de la base por centrales se detalla todas y cada una de las centrales tipo ERICSSON existentes así como su descripción y el número total de las mismas.

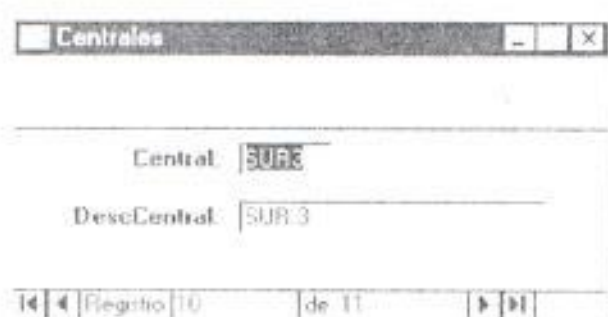


Fig. 4.32

Es así que el número total de las centrales es de 10 por lo que si se desea ingresar alguna otra se deberá acceder al número 11.

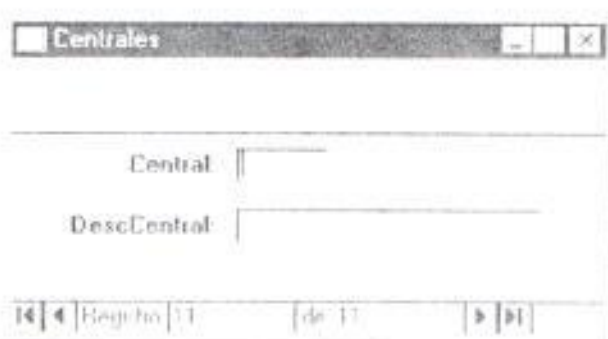


Fig. 4.33

Una vez que se ubica en el número 11 se percatará que está en blanco y con el cursor en el recuadro de central.

Centrales

Central:

DescCentral:

14 | < | Registro | 11 | de 11 | > | >> |

Fig. 4.34

Es aquí donde se deberá digitar el nombre de la nueva central así como de su descripción.

4.5.3 SELECCION DE BASE POR RUTAS

Emetel Rutas

Ruta:

Descripción:

TipoRuta:

Grupo:

Central:

14 | < | Registro | 1 | de 300 | > | >> |

Fig. 4.35

En esta selección se detalla todas las rutas existentes con los siguientes parámetros: descripción de la ruta, tipo de ruta (entrante o saliente), grupo (regional, internacional, etc.) y la central a la cual está conectada.

Emetel Rutas

Ruta: 860

Descripción: URDESAT

TipoRuta: 0

Grupo: AS

Central: BELL

« « Regreso 860 de 860 » »

Fig. 4.36

Es así que llegamos a la última ruta que en este caso es la 860. Luego nos percatamos que para poder incluir una nueva ruta deberemos ubicarnos en la siguiente posición:

Emetel Rutas

Ruta: _____

Descripción: _____

TipoRuta: _____

Grupo: _____

Central: _____

« « Regreso 861 de 861 » »

Fig. 4.37

Entonces aparecerá una ventana sin ningún dato y con el cursor en la parte superior.

Emetel Rutas

Ruta: NUEVA

Descripción: NUEVA

TipoRuta: F

Grupo: RE

Central: TR56

Registros: 1 de 1

Fig. 4.38

Ahora se ingresarán los datos de la ruta nueva, para lo cual deberemos especificar con seguridad todas sus características.

4.5.4 SELECCION DE BASE POR GRUPO DE CENTRALES

Emetel - Grupo

Grupo: RE

DescGrupo: CENTRALES ENTRANTES

Registros: 1 de 13

Fig. 4.39

En la base de grupo de centrales se detallan todos los tipos modelos de centrales que existen y que sirven para realizar la subdivisión de los informes.

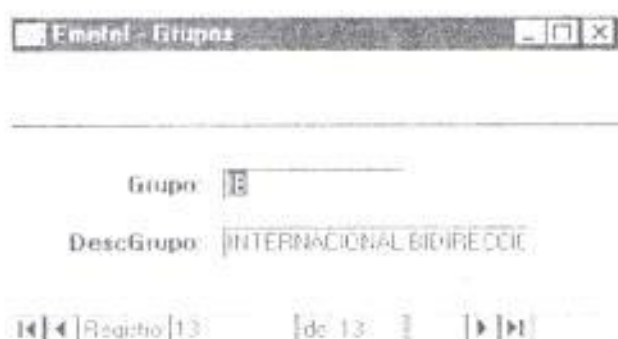


Fig. 4.40

Como en las anteriores bases de datos, se puede llegar al último grupo de centrales.

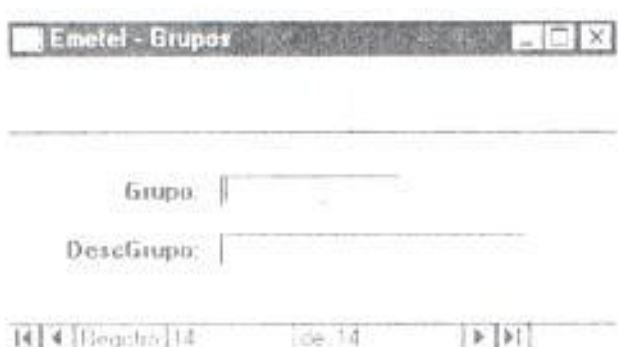


Fig. 4.41

En la siguiente posición encontraremos una ventana sin datos con el cursor en la cuadrícula superior.

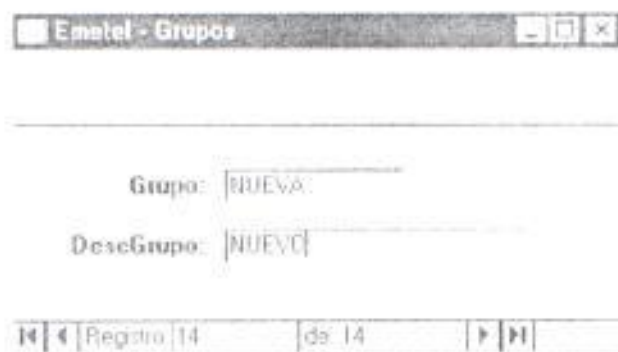


Fig. 4.42

Es entonces cuando se deberán ingresar los datos del nuevo grupo de centrales que se hayan creado.

4.5.5 ACTUALIZACION DE LA BASE DE DATOS

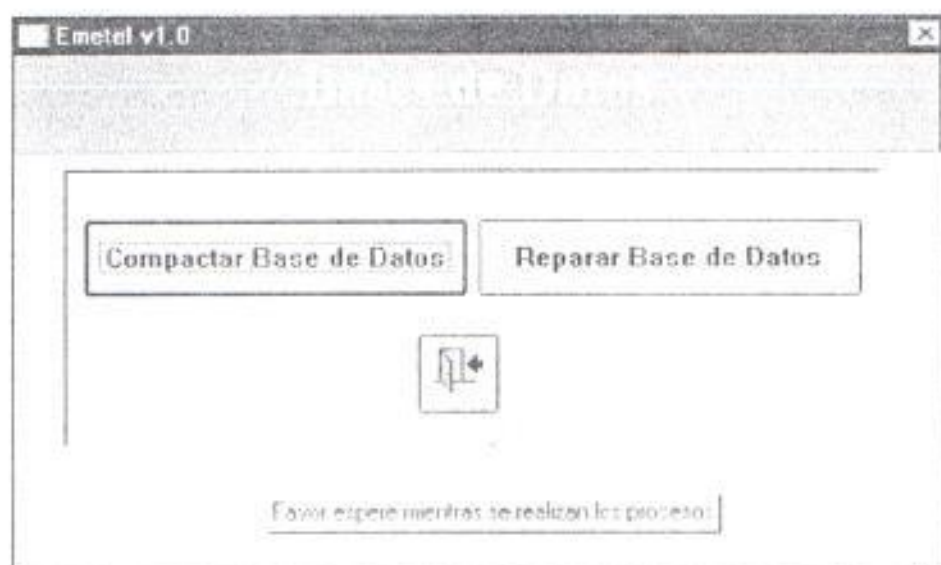


Fig. 4.43

En este icono se tendrá la opción de realizar la reparación de la base de datos una vez que se haya actualizado los datos en las respectivas bases al final de las mismas. Ese procedimiento ya se lo explicó en las páginas anteriores, por lo que al final solo queda presionar la opción "REPARAR BASE DE DATOS".



Fig. 4.44

Cuando aparece esta ventana, se indica que la base ha sido reparada por lo que ya se podrá ingresar datos en esas nuevas centrales, rutas, etc. para que sean procesadas.

CONCLUSIONES Y RECOMENDACIONES

La elaboración del presente proyecto no tendría ningún objetivo si es que al final nuestro grupo no hubiera sido capaz de llegar a un consenso y emitir nuestra propia visión madura de lo que hemos aprendido, es decir nuestras propias conclusiones.

Es así que en el transcurso de estas semanas de arduo trabajo llegamos a formar una idea concisa acerca de diversos puntos que a continuación detallaremos:

En primer lugar podemos mencionar lo importante que es para un profesional tener una visión clara de la realidad en la cual trabaja, ya que con esta visión podremos ser capaces de tomar decisiones acertadas para poder mejorar tanto profesional como personalmente. Es así que en el departamento de calidad de servicio de EMETEL nos percatamos que se llevaban a cabo diversas tareas de elaboración de informes tales como tráfico de llamadas, calidad de servicio y carga del procesador, los cuales muchas veces eran modificados por los operadores de la central con datos equivocados, por lo que estos informes no tenían un peso real y por consiguiente brindaban una referencia hasta cierto punto errada para la toma de futuras decisiones, por Ej. ampliación de circuitos, etc.

Es entonces cuando decidimos que la única forma de llevar a cabo el presente proyecto fue aprender todo el funcionamiento de la obtención de datos de mediciones del sistema AXE y el tipo de informe final que se necesitaba en base a dichos datos.

Con estos datos elaboramos diversos procedimientos para el procesamiento correcto de los mismos de tal forma que el informe final no tenga que pasar por las manos de ningún operador, por lo que se aseguró que este informe sea lo más real posible.

Otra conclusión es lo importante que es aprender computación hoy en día para poder ser un buen profesional, ya que el conocimiento de algún lenguaje de programación y de utilitarios ha sido vital en el presente trabajo.

El manejo de los datos fue realizado en ACCESS 2.0 y la presentación de los informes en VISUAL BASIC 3.0. El procesamiento de los datos fue realizado en algunas rutinas de Lenguaje C.

El manejo del programa realizado por nuestro grupo es sumamente fácil ya que se implementó un sistema de ventanas y menús en Visual Basic por lo que el operador no tendrá mayores problemas en el manejo del mismo.

En lo que respecta al procesamiento de los datos de tráfico en Erlangs, se usó los conocimientos de método numéricos para obtener el número de circuitos ideales por cada ruta. Para esto se usaron aproximaciones sucesivas en la fórmula de Erlang con la opción de utilizar el 1% ó el 2% con un error de aproximación de una milésima.

El tipo de gráfico que se usa para la elaboración de los informes de tráfico sean estos por ruta o de hora pico es en barras 3D, en los cuales se incluye el parámetro de circuitos ideales por lo que se podrá comparar con la cantidad de circuitos existentes para poder tomar decisiones sobre futuras ampliaciones o establecer si existe un sobredimensionamiento en alguna ruta.

Para los gráficos de calidad de servicio se usan dos tipos, ya sean en forma pastel 3D o en barras 2D. En el caso de pastel 3D se incluyen los datos de calidad de servicio por central, sacando un porcentaje general de todas las rutas salientes que se conectan a dicha central. Para el gráfico de barra 2D se incluyen los datos de calidad de servicio de todas y cada una de las rutas salientes conectadas a dicha central.

La forma de gráfico que se usa en carga del procesador y de tráfico es de tipo lineal. En estos se incluyen valores de la carga por hora que ha recibido el procesador de la central.

Una recomendación muy importante es aquella que si los gráficos requieren ser modificados, ya sea por anexar fecha o comentarios, se recomienda el uso de un programa de captura como lo es el CORAL DRAW cualquier versión.

BIBLIOGRAFIA

- ♦ **ERICSSON:** Seminario sobre mediciones en AXE

- ♦ **TRAFICO TELEFONICO:** INICTEL Instituto Nacional de Investigación y Capacitación de Telecomunicaciones. Elaborado por : Ings. Zenón Choque, Bernardo Castro, Luis Rodríguez, Rodolfo Vargas, Carlos Mejía.

- ♦ **TABLAS DE ERLANGS:** Manual propiedad de EMETEL.

ANEXO A

CARACTERISTICAS DE HARDWARE Y SOFTWARE

Para que el programa "Procesador Gráfico de Datos" pueda ejecutarse satisfactoriamente, el equipo en que se instale debe satisfacer las siguientes características:

COMPUTADORA	La computadora debe ser IBM o compatible, con un microprocesador de la familia INTEL 486DX2 o superior.
MEMORIA RAM	La computadora debe poseer al menos 8 Mega Bytes de Memoria Ram
MEMORIA EN DISCO	Este programa necesita para si mismo un minimo de 10 Mega Bytes de espacio disponible, pero dado que constantemente se esta generando nueva información, se recomienda 20 Mega Bytes de espacio disponible.
PUERTOS	Deben existir puertos seriales y paralelos disponibles para la recepción de datos y conexiones con la impresora
IMPRESORAS	Opcionalmente puede existir una impresora conectada para imprimir gráficos o archivos.
SISTEMA OPERATIVO	Se requiere que la computadora posea sistema operativo de ambiente gráfico multitarea tal como Windows 3.1 o superior.



VELOCIDAD DE LA MAQUINA	Se requiere que la velocidad de procesamiento de la PC sea al menos de 66 Mega Hertz.
MODEMS	Dado que el programa se conecta en red, la computadora debe poseer un modem instalado, preferiblemente de alta velocidad.

PROCESO DE INSTALACION DEL PROGRAMA

Para instalar el programa deben seguirse los procedimientos clásicos de instalación de un programa para Windows, básicamente deben seguirse los siguientes pasos:

- Inserte el disco de instalación titulado DISK ½ en su unidad de disco flexible.
- Si no lo ha hecho, ingrese a Windows.
- Desde el manejador de archivos seleccione el drive del disco, posicione en el archivo SETUP.EXE y ejecútelo haciéndole doble click sobre el o desde una opción de menú.
- Después de inicializado el proceso de instalación, aparecerá la siguiente pantalla donde se solicitará el nombre del directorio donde se instalará el programa, la opción por default es C:\EMETEL2

RESTRICCIONES DEL PROGRAMA

- * Se recomienda que el computador en el cual se instale el administrador, posea los siguientes programas: **VISUAL BASIC 3.0**, **ACCESS 2.0** y la compatibilidad entre ambos.
- * Al cargar un archivo en el cual existan nuevas rutas las cuales no constan en la base de datos del programa, estas serán eliminadas y no aparecerán

en el reporte seleccionado. Por lo tanto es imprescindible, la actualización de dicha base de datos antes de la operación del programa.

- * Cuando se actualice la base de datos al aumentar una nueva central, esta no deberá tener más de cuatro caracteres; y al aumentar una nueva ruta no deberá tener más de ocho caracteres.
- * Dependiendo del tipo de informe que se requiera y la opción seleccionada (carga, tráfico y calidad) el archivo que se cargue deberá corresponder a dicha opción caso contrario el programa al leer parámetros diferentes presentará errores y abortará la ejecución.
- * El nombre de la central seleccionada al "bajar" información de red o trabajar con archivos deberá tener el mismo formato del programa: **TRSG, BELL, QUEV, MACH, NOR2, ALB2, MAPA, CEN3, SUR2, DURA.**
- * Es responsabilidad de la persona que ingrese el archivo saber cual es la central , fecha y hora de la cual se requiere reporte, debido a que el programa solicita estos datos; si se ingresan datos diferentes a los que constan en el archivo el programa presenta el siguiente mensaje : " No hay datos para graficar ". Una vez cargado el archivo hay posibilidad de visualizar información como horas y fechas del archivo en uso, pero no de centrales, entrante a la carpeta **TMP** del directorio **EMETEL**.

ANEXO B

DEFINICIONES GENERALES.

CRONOGRAMA.

El cronograma se usa para especificar cuando una medición en particular comenzará, la duración de la medición, la cantidad de mediciones diarias y los días en que se harán las mediciones. Un cronograma esta ligado al programa de medición definido para una medición en particular.

HORA PICO.

Es la hora del día que la central transportara la mayor cantidad de tráfico.

ERLANG.

Es la unidad que se usa para medir la intensidad de tráfico telefónico. Un erlang puede ser definido como la ocupación de un circuito durante una hora.

$$\text{ERLANG} = \frac{\text{Intentos} \times \text{Tiempo de Ocupación}}{3600} - \text{Congestión}$$

SISTEMA DE COLA.

En una ruta con sistema de cola si todos los dispositivos están ocupados, el siguiente intento entrara en cola y cuando se libere un dispositivo la llamada podrá proseguir. Sin embargo, si la cola esta llena los intentos fracasarán.

SISTEMA DE PERDIDA.

Si en una ruta con sistema de pérdida todos los dispositivos están ocupados, la siguiente llamada entrará en congestión y fracasará. Las rutas BT trabajan con sistema de pérdida.

TRD (DESTINO DE TRÁFICO).

Es un parámetro que se define en las tablas de análisis de número B para identificar determinados destinos para la medición de dispersión de tráfico.

TRDG (GRUPO DE DESTINO DE TRÁFICO).

Es una agrupación de varios TRDs que se usa para la medición de dispersión de tráfico.

TRG (GRUPO DE REGISTRO DE TRÁFICO).

Es una agrupación de rutas que se usa para medición de tráfico.

CONCEPTOS DE TRÁFICO TELEFONICO**ESTADO DE UN ORGANO O CIRCUITO.**

Un órgano o circuito de una central de Conmutación Telefónica en perfecto funcionamiento, cuando es observado a lo largo del tiempo puede ser representado bajo dos aspectos en cuanto a su estado:

a.- **Libre:** un órgano o circuito se dice "Libre" cuando esta disponible para ser utilizado por cualquier llamada que lo solicite.

b.- **Ocupado:** un órgano o circuito se dice "ocupado" cuando esta siendo utilizado por una llamada telefónica en curso.

La fig. 1 ilustra los estados de un órgano o circuito a lo largo del tiempo.

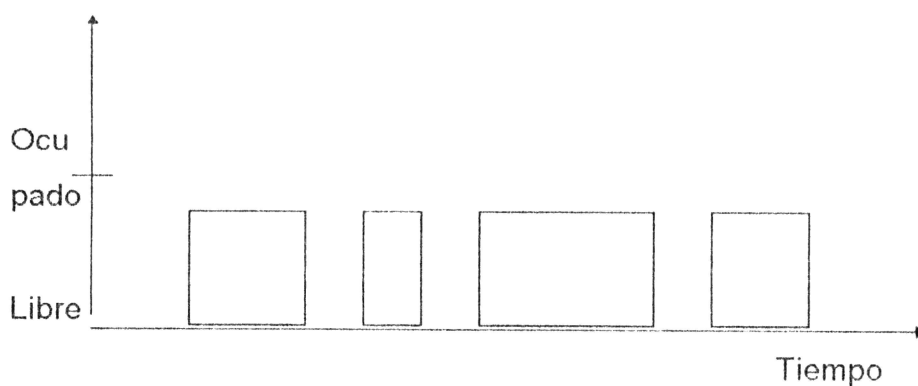


Fig. 1

TOMA DE UN ORGAN O CIRCUITO.

Es el paso, del órgano o circuito, del estado libre al estado ocupado.

REGISTRO DE OCUPACIONES DE ORGANOS O CIRCUITOS.

A.- Registro de Ocupaciones Individuales:

Es el método utilizado para registrar las ocupaciones de un grupo de órganos o circuitos, a través de registros de las ocupaciones de cada órgano o circuito individualmente.

La fig. 2 ilustra las ocupaciones de un grupo de 5 circuitos a través del registro de ocupaciones individuales.

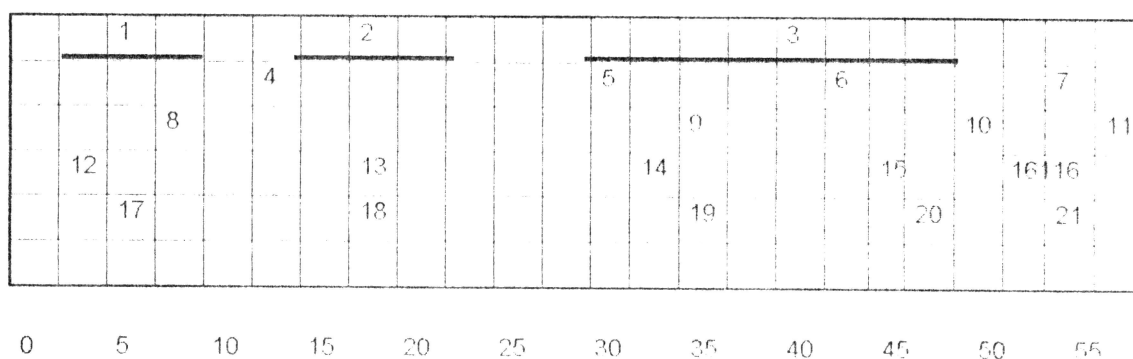


Fig. 2

B.- Registro de Ocupaciones Simultáneas:

Es el método utilizado para registrar las ocupaciones de un grupo de órganos o circuitos a través del registro de las ocupaciones que ocurren simultáneamente.

La **fig. 3** ilustra los datos mostrados en la **figura 2** a través del registro de las ocupaciones simultáneas.

Número de circuitos ocupados

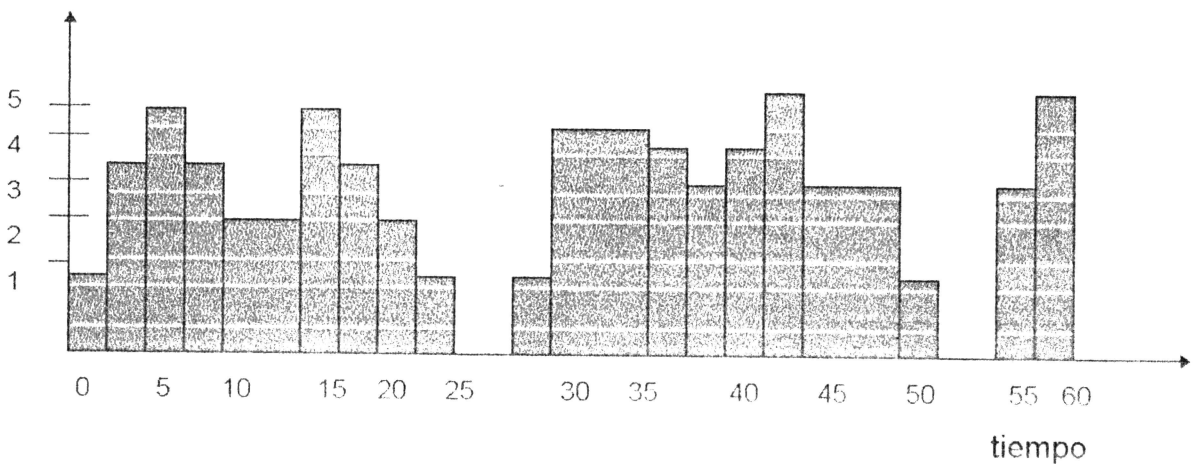


Fig. 3

TIEMPO DE OCUPACION (t_j)

Es el tiempo durante el cual una llamada telefónica ocupa un órgano o circuito de la central.

PERIODO DE OBSERVACION (T).

Es el tiempo durante el cual un órgano o circuito, grupos de órganos o circuitos son supervisados.

VOLUMEN DE TRÁFICO (V).

Es la sumatoria de los tiempos de ocupación de un órgano o circuito o grupos de órganos o circuitos de la central durante el periodo de observación.

El volumen de tráfico se calcula a través de la fórmula 1.

$$V = \sum_{j=1}^n t_j \quad (1)$$

donde: n = número total de tomas

t_j = tiempo de ocupación de la j -ésima toma

La unidad del volumen de tráfico es el "tiempo", en tanto, para indicar que la misma se refiere a "Tráfico Telefónico" se utiliza la unidad de tiempo seguido de la palabra "Erlang".

"El volumen de tráfico de un grupo de órganos es igual a la sumatoria del volumen de tráfico de cada órgano individualmente".

TASA DE TOMAS.

Es el número de tomas de un órgano o circuito, o grupo de órganos o circuitos por periodo de observación.

La tasa de tomas se calcula a través de la fórmula 2.

$$i = n / T \quad (2)$$

donde : n = número de tomas

T = periodo de observación

La tasa de tomas se expresa en tomas por unidad de tiempo.

TIEMPO MEDIO DE OCUPACION.

Es la media aritmética de los tiempos de ocupación de un órgano o circuitos o un grupo de órganos o circuitos de la Central de Conmutación, durante el periodo de observación.

El tiempo medio de ocupación se calcula a través de la fórmula 3.

$$t = (\sum_{j=1}^n t_j) / n \quad (3)$$

El tiempo medio de ocupación puede también calcularse a través de la relación entre el volumen de tráfico (V) y el número total de tomas (n) durante el periodo de observación, fórmula 4.

$$t = V / n \quad (4)$$

El tiempo medio de ocupación se expresa en unidades de tiempo por toma.

INTENSIDAD DE TRÁFICO.

La intensidad de tráfico es una cantidad adimensional Internacionalmente se expresa en "ERLANG", nombre dado en homenaje al Dinamarqués A. K. Erlang por sus notables contribuciones a la teoría de tráfico telefónico.

Un Erlang representa un órgano ocupado durante todo el periodo de observación.

La intensidad de tráfico representa:

- a) El número medio de llamadas cursadas simultáneamente durante el periodo de observación.

- b) El número medio de circuitos ocupados simultáneamente durante el periodo de observación.
- c) El número medio de llamadas originadas durante un periodo de observación igual al tiempo medio de duración de las llamadas.

INTENSIDAD DE TRÁFICO OFRECIDO (A).

La intensidad de tráfico ofrecido puede definirse como el número medio de llamadas originadas durante un periodo de observación igual al tiempo medio de duración de las llamadas.

La intensidad de tráfico ofrecido no es medible. Su valor se obtiene a través de estimados.

INTENSIDAD DE TRÁFICO CURSADO (A_c).

La intensidad de tráfico cursado se define como el número medio de ocupaciones simultáneas ocurridas en el periodo de observación.

La intensidad de tráfico cursado puede obtenerse a través de mediciones.

Puede calcularse también a través de la relación entre el volumen de tráfico (V) y el tiempo de observación (T). Fórmula 5.

$$A_c = V / T \quad (5)$$

La intensidad de tráfico cursado puede también calcularse por el producto de la tasa de tomas (i) por el tiempo de ocupación (t). Fórmula 6.

$$A_c = i \cdot t \quad (6)$$

CONGESTIONAMIENTO.

Es el estado del sistema telefónico caracterizado por la ocupación de todos los medios de conexión.

El congestionamiento puede ser considerado bajo dos aspectos distintos definidos:

a) Congestionamiento en la Llamada (B).

El congestionamiento en la llamada es el porcentaje de las llamadas que encuentran todos los medios de conexión ocupados.

El congestionamiento en la llamada se calcula a través de la fórmula 7.

$$B = P / N \quad (7)$$

donde: **B** = congestionamiento en la llamada.

P = número de tentativas de llamadas que encuentran todos los medios de conexión ocupados.

N = número total de tentativas de llamadas.

b) Congestionamiento en el tiempo (E).

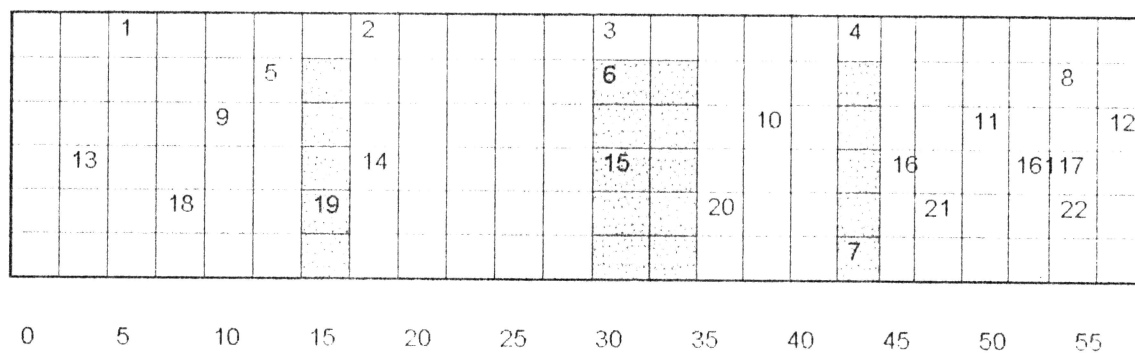
Es la proporción del tiempo, en relación al periodo de observación, en que todos los órganos o circuitos permanecen ocupados simultáneamente.

El congestionamiento en el tiempo se calcula a través de la fórmula 8.

$$E = t / T \quad (8)$$

donde: t = sumatoria de los periodos en que todos los órganos o circuitos se encuentran ocupados.

T = tiempo de observación.



Tiempo en Minutos

Fig. 4

CIRCUITOS IDEALES

Circuitos que idealmente debe tener una central para satisfacer el tráfico requerido.

TRAFICO OFRECIDO

Se puede definir como el número promedio de llamadas originadas durante un periodo de observación.

CIRCUITOS EXISTENTES

Es la cantidad de circuitos instaladas en la central.

ANEXO C

CODIGOS DE SUBRUTINAS

RECIBIR COLOR DE DATOS

```

Sub Recibir_ColorData (ByVal puntos As Integer, ByVal setes As Integer, x As Form)
    Dim i As Integer
    For i = 1 To puntos
        x.Combo1.AddItem Str$(i)
    Next i
    For i = 1 To setes
        x.Combo2.AddItem Str$(i)
    Next i
End Sub

```

ACTUALIZAR LISTA

```

Dim qs_i As Integer
    Dim cod_error As Integer
    Dim dyn_lista As Dynaset

    On Error GoTo err_ActLista
    'Se crea el query para obtener los datos
    Set dyn_lista = mibase.CreateDynaset(tabla)
    ualista.Clear ' se limpia la lista
    If dyn_lista.RecordCount > 0 Then
        'Si encuentra registro procede a insertarlo en la lista
        qs_i = 0
        Do While dyn_lista.EOF = False
            ualista.AddItem Trim$(dyn_lista.Fields(descripcion))
            ualista.ItemData(qs_i) = dyn_lista.Fields(codigo)
            dyn_lista.MoveNext
            qs_i = qs_i + 1
        Loop
    End If

```

```

    Loop
    ughosta.ListIndex = -1
End If
'ugghosta.listcount
' tresp.Text = ""
Exit Sub

err_ActLista:
    MsgBox "Error: " + CStr(Err) + " " + Error
Exit Sub

End Sub

```

FORMATO TEXTO

```

Sub Format_Texto_Tmp (arch_fuente As String)
    Dim rc As Integer
    Dim X, Z%
    Dim str_comando As String
    Dim fuentes As String * 50
    Dim destino As String * 50
    Dim PROBABILIDAD As String * 50

    On Error GoTo err_Format_Texto_Tmp
    'str_comando = DirBin + "TRAFICO.EXE " + DirDatos + "trafalb2 l02 " + DirTmp +
    "Trafico tmp .01"

    If bdOpcion = 0 Then
        str_comando = DirBin + "CALIDAD EXE " + arch_fuente + " " + DirTmp + "Calidad tmp"
    Else
        str_comando = DirBin + "TRAFICO.EXE " + arch_fuente + " " + DirTmp + "Trafico tmp "
+ PROBABILIDAD1
    End If
    X = Shell(str_comando, 1)
    While GetModuleUsage(X) > 0

```

```

        Z% = DoEvents()
    Wend
    Exit Sub
err_Format_Texto_Tmp:
    Resume Next
End Sub

```

ITEM LISTA

```

Function get_item_lista (ugcontrol As Control, str_lab As String, band As Integer) As String
Dim qs_i As Integer
On Error GoTo err_itemlst
    Select Case band
        Case 1 ' Devuelvo codigo en string comparando con el nombre
            For qs_i = 0 To ugcontrol.ListCount - 1
                If str_lab = Trim$(ugcontrol.List(qs_i)) Then
                    get_item_lista = CStr(ugcontrol.ItemData(qs_i))
                    ugcontrol.ListIndex = qs_i
                    Exit For
                End If
            Next qs_i
        Case 2 ' devuelvo el nombre comprando con el codigo
            For qs_i = 0 To ugcontrol.ListCount - 1
                If CInt(str_lab) = ugcontrol.ItemData(qs_i) Then
                    ugcontrol.ListIndex = qs_i
                    get_item_lista = ugcontrol.List(qs_i)
                    Exit For
                End If
            Next qs_i
        Case 3 ' devuelvo el codigo comprando con el codigo
            For qs_i = 0 To ugcontrol.ListCount - 1
                If CInt(str_lab) = ugcontrol.ItemData(qs_i) Then
                    ugcontrol.ListIndex = qs_i
                    get_item_lista = ugcontrol.ItemData(qs_i)
                    Exit For
                End If
            Next qs_i
    End Select

```

```

        End If
    Next qs_i

    End Select
Exit Function
err_itemlst:
    MsgBox "Error:"
End Function

```

LLENAR REGISTRO

```

Rem Llena un lista o un combo desde una tabla segun un criterio'
Sub llena_listaRegistro (criterio As String, ugcontrol As Control, codigo As String,
descripcion As String)
Dim mitabla As Dynaset
Dim qs_i As Integer
Dim qx_lon As Long

On Error GoTo err_llenalst

Set mitabla = mibase.CreateDynaset(criterio)

qs_i = 0
ugcontrol.Clear
If mitabla.RecordCount > 0 Then
    Do Until mitabla.EOF
        ugcontrol.AddItem Trim$(mitabla.Fields(codigo)) + " - " +
Trim$(mitabla.Fields(descripcion)), qs_i
        'ugcontrol.ItemData(qs_i) = mitabla.Fields(codigo)
        mitabla.MoveNext
        qs_i = qs_i + 1
    Loop
ugcontrol.ListIndex = -1
End If

```



```

Exit Sub
err_llenalst:
    MsgBox "Error." + Str(Err) + Error(Err)

End Sub

```

FORMATO DE LINEA

```

Function Formato_linea (x As String) As String
    Dim i As Integer
    For i = Len(x) To 15
        x = x + " "
    Next i
    Formato_linea = x
End Function

```

GRAFICAR

```

Sub Graficar_calidad ()
    Dim Path As String
    Dim temp As calidad_reg
    Dim titulo As String, Fecha As String
    Dim linea As String
    Dim prere As Integer, bnans As Integer, bbusy As Integer
    Dim bans As Integer, tefault As Integer, fulfil As Integer
    Path = App.Path
    If Right$(Path, 1) <> "\" Then Path = Path + "\"
    Path = Path + "temp.txt"
    flag_cancelar = False
    Open Path For Input Shared As #2
        Line Input #2, linea
        Line Input #2, linea
        Fecha = linea
    Do While Not EOF(2) And flag_cancelar = False
        Line Input #2, linea
    
```

```

Line Input #2, linea
titulo = "Calidad de Servicio " + linea + " Fecha " + Fecha
Line Input #2, linea
Line Input #2, linea
Line Input #2, linea
Line Input #2, linea
prere = Val(linea)
Line Input #2, linea
Line Input #2, linea
bnans = Val(linea)
Line Input #2, linea
Line Input #2, linea
bans = Val(linea)
Line Input #2, linea
Line Input #2, linea
bbusy = Val(linea)
Line Input #2, linea
Line Input #2, linea
tefault = Val(linea)
fulfil = prere + bnans + bbusy + bans + tefault
Load FRM_Graficar
FRM_Graficar.Graph1.GraphTitle = titulo
FRM_Graficar.Graph1.NumPoints = 5
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.GraphData = prere
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.LabelText = "A-REPONE " + Str$((prere * 100) / fulfil) + "%"
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.GraphData = bnans
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.LabelText = "B-NO-CONTESTA " + Str$((bnans * 100) / fulfil) +
"%"
FRM_Graficar.Graph1.ThisPoint = 3
FRM_Graficar.Graph1.GraphData = bbusy
FRM_Graficar.Graph1.ThisPoint = 3

```

```

FRM_Graficar.Graph1.LabelText = "B-OCUPADO " + Str$((bbusy * 100) / fulfil) + "%"
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.GraphData = bans
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.LabelText = "B-CONTESTA " + Str$((bans * 100) / fulfil) + "%"
FRM_Graficar.Graph1.ThisPoint = 5
FRM_Graficar.Graph1.GraphData = tefault
FRM_Graficar.Graph1.ThisPoint = 5
FRM_Graficar.Graph1.LabelText = "FALLA TECNICA " + Str$((tefault * 100) / fulfil) + "%"
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.LegendText = "A-Repone"
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.LegendText = "B no contesta"
FRM_Graficar.Graph1.ThisPoint = 3
FRM_Graficar.Graph1.LegendText = "B ocupado"
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.LegendText = "B contesta"
FRM_Graficar.Graph1.ThisPoint = 5
FRM_Graficar.Graph1.LegendText = "Falla Tecnica"
FRM_Graficar.Show 1
Loop
Close #2

```

GRAFICAR 2

```

Sub Graficar_calidad2 ()
Dim Path As String
Dim temp As calidad_reg
Dim titulo As String, Fecha As String
Dim linea As String, contador As Integer, x As Integer

Dim prere As Integer, bnans As Integer, bbusy As Integer
Dim bans As Integer, tefault As Integer, fulfil As Integer
Path = App.Path
If Right$(Path, 1) <> "\" Then Path = Path + "\"

```

```

Path = Path + "temp.txt"
flag_cancelar = False
Open Path For Input Shared As #2
    Line Input #2, linea
    Line Input #2, linea
    Fecha = linea
    titulo = "Calidad de Servicio Fecha " + Fecha
    contador = 1
    Do While Not EOF(2) And flag_cancelar = False
        Line Input #2, linea
        contador = contador + 1
    Loop
Close
contador = contador / 14
x = contador
Load FRM_Graficar

FRM_Graficar.Graph1.NumPoints = contador
FRM_Graficar.Graph1.NumSets = 5
FRM_Graficar.Graph1.GraphStyle = 2
FRM_Graficar.Graph1.GraphTitle = titulo
Open Path For Input Shared As #2
    Line Input #2, linea
    Line Input #2, linea
    For contador = 1 To x
        Line Input #2, linea
        Line Input #2, linea
        FRM_Graficar.Graph1.ThisPoint = contador
        FRM_Graficar.Graph1.LabelText = linea
        Line Input #2, linea
        Line Input #2, linea

        Line Input #2, linea
        Line Input #2, linea
        prere = Val(linea)

```

```
Line Input #2, linea
Line Input #2, linea
bnans = Val(linea)
Line Input #2, linea
Line Input #2, linea
bans = Val(linea)
Line Input #2, linea
Line Input #2, linea
bbusy = Val(linea)
Line Input #2, linea
Line Input #2, linea
tefault = Val(linea)
fulfil = prere + bnans + bbusy + bans + tefault
FRM_Graficar.Graph1.ThisPoint = contador
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.GraphData = (prere * 100) / fulfil
FRM_Graficar.Graph1.ThisPoint = contador
FRM_Graficar.Graph1.ThisSet = 2
FRM_Graficar.Graph1.GraphData = (bnans * 100) / fulfil
FRM_Graficar.Graph1.ThisPoint = contador
FRM_Graficar Graph1.ThisSet = 3
FRM_Graficar.Graph1.GraphData = (bans * 100) / fulfil
FRM_Graficar.Graph1.ThisPoint = contador
FRM_Graficar.Graph1.ThisSet = 4
FRM_Graficar.Graph1.GraphData = (bbusy * 100) / fulfil
FRM_Graficar Graph1 ThisPoint = contador
FRM_Graficar Graph1.ThisSet = 5
FRM_Graficar Graph1.GraphData = (tefault * 100) / fulfil
Next contador
```

Close

```
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.LegendText = "A Reponse"
FRM_Graficar.Graph1.ThisPoint = 2
```

```

FRM_Graficar.Graph1.ThisSet = 2
FRM_Graficar.Graph1.LegendText = "B no contesta"
FRM_Graficar.Graph1.ThisPoint = 3
FRM_Graficar.Graph1.ThisSet = 3
FRM_Graficar.Graph1.LegendText = "B ocupado"
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 4
FRM_Graficar.Graph1.LegendText = "B contesta"
FRM_Graficar.Graph1.ThisPoint = 5
FRM_Graficar.Graph1.ThisSet = 5
FRM_Graficar.Graph1.LegendText = "Falla Tecnica"
FRM_Graficar.Show 1
End Sub

```

GRAFICAR CARGA

```

Sub Graficar_Carga ()
Dim Path As String
Dim titulo As String
Dim linea As String
Dim contador As Integer, x As Integer
Dim tiempo As String
Dim ftchdo As Integer, ftchdi As Integer
Dim llamadas As Integer
Dim temp1 As Single, temp2 As Single
Dim Fecha As String
contador = 1
flag_cancelar = False
Path = App.Path
If Right$(Path, 1) <> "\" Then Path = Path + "\"
Path = Path + "temp.txt"
Open Path For Input Shared As #2
    Line Input #2, linea
    Line Input #2, linea
    Fecha = linea

```

```
Do While Not EOF(2)
  Line Input #2, linea
  contador = contador + 1
Loop
Close
contador = contador / 12
x = contador
  titulo = "Carga del Procesador Fecha " + Fecha
  Load FRM_Graficar
  FRM_Graficar.Graph1.GraphTitle = titulo
  FRM_Graficar.Graph1.NumPoints = contador

Open Path For Input Shared As #2
  contador = 1
  Line Input #2, linea
  Line Input #2, linea

Do While Not EOF(2)
  Line Input #2, linea
  Line Input #2, linea
  tiempo = linea
  Line Input #2, linea
  Line Input #2, linea
  llamadas = Val(linea)
  Line Input #2, linea
  Line Input #2, linea

  Line Input #2, linea
  Line Input #2, linea

  Line Input #2, linea
  Line Input #2, linea

  Line Input #2, linea
  Line Input #2, linea
```

```

FRM_Graficar.Graph1.ThisPoint = contador
FRM_Graficar.Graph1.GraphData = llamadas
If Val(tiempo) Mod 100 = 0 Then
FRM_Graficar.Graph1.ThisPoint = contador
FRM_Graficar.Graph1.LabelText = tiempo
End If
    contador = contador + 1
Loop
Close #2
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.LegendText = "Pload"
FRM_Graficar.Show 1

If flag_cancelar = False Then
    FRM_Graficar.Graph1.DataReset = 9
    titulo = "Carga del Procesador"
    FRM_Graficar.Graph1.GraphTitle = titulo
    FRM_Graficar.Graph1.NumPoints = x
    FRM_Graficar.Graph1.NumSets = 2
Open Path For Input Shared As #2
    contador = 1
    Line Input #2, linea
    Line Input #2, linea

Do While Not EOF(2) And flag_cancelar = False
    Line Input #2, linea
    Line Input #2, linea
    tiempo = linea
    Line Input #2, linea
    Line Input #2, linea
    llamadas = Val(linea)
    Line Input #2, linea
    Line Input #2, linea
    temp1 = Val(linea)

```



```
Line Input #2, linea
Line Input #2, linea
temp1 = (temp1 + Val(linea)) * 3600
Line Input #2, linea
Line Input #2, linea
temp2 = Val(linea)
Line Input #2, linea
Line Input #2, linea
temp2 = (temp2 + Val(linea)) * 3600

FRM_Graficar.Graph1.ThisPoint = contador
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.GraphData = temp1
FRM_Graficar.Graph1.ThisPoint = contador
FRM_Graficar.Graph1.ThisSet = 2
FRM_Graficar.Graph1.GraphData = temp2
If Val(tiempo) Mod 100 = 0 Then
FRM_Graficar.Graph1.ThisPoint = contador
FRM_Graficar.Graph1.LabelText = tiempo
End If
    contador = contador + 1
Loop
Close #2
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.LegendText = "BHC"
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.ThisSet = 2
FRM_Graficar.Graph1.LegendText = "BHCA"
FRM_Graficar.Show 1

End If
End Sub
```

GRAFICAR PICOS

```

Sub Graficar_Picos ()
  Dim Path As String, Fecha As String
  Dim linea As String, titulo As String
  Dim x1 As Single, x2 As Single
  Path = App.Path
  flag_cancelar = False
  If Right$(Path, 1) <> "\" Then Path = Path + "\"
  Path = Path + "temp2.txt"
  Open Path For Input Shared As #4
  Line Input #4, linea
  Line Input #4, linea
  Fecha = linea
  titulo = "Trafico por Rutas (horas pico) Fecha " + Fecha
  Load FRM_Graficar

  Do While Not EOF(4) And flag_cancelar = False
    On Error GoTo final1:
    FRM_Graficar.Graph1.GraphTitle = titulo
    FRM_Graficar.Graph1.DataReset = 9
    FRM_Graficar.Graph1.NumPoints = 4
    FRM_Graficar.Graph1.NumSets = 4
    FRM_Graficar.Graph1.ThisPoint = 1
    FRM_Graficar.Graph1.ThisSet = 1
    FRM_Graficar.Graph1.LegendText = "Existente"
    FRM_Graficar.Graph1.ThisPoint = 2
    FRM_Graficar.Graph1.ThisSet = 2
    FRM_Graficar.Graph1.LegendText = "Ideal"
    FRM_Graficar.Graph1.ThisPoint = 3
    FRM_Graficar.Graph1.ThisSet = 3
    FRM_Graficar.Graph1.LegendText = "Ofrecido"
    FRM_Graficar.Graph1.ThisPoint = 4
    FRM_Graficar.Graph1.ThisSet = 4
    FRM_Graficar.Graph1.LegendText = "Diferencia"
  
```

Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.LabelText = linea
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.GraphData = Val(linea)
x1 = Val(linea)

Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.ThisSet = 2
FRM_Graficar.Graph1.GraphData = Val(linea)
x2 = Val(linea)

Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.ThisSet = 3
FRM_Graficar.Graph1.GraphData = Val(linea)

FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.ThisSet = 4
FRM_Graficar.Graph1.GraphData = Abs(x1 - x2)

#2

Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.ThisSet = 1

```

FRM_Graficar.Graph1.LabelText = linea
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.GraphData = Val(linea)
x1 = Val(linea)
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.ThisSet = 2
FRM_Graficar.Graph1.GraphData = Val(linea)
x2 = Val(linea)
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.ThisSet = 3
FRM_Graficar.Graph1.GraphData = Val(linea)

FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.ThisSet = 4
FRM_Graficar.Graph1.GraphData = Abs(x1 - x2)

'#3
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 3
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.LabelText = linea
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 3
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.GraphData = Val(linea)
x1 = Val(linea)

```

Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 3
FRM_Graficar.Graph1.ThisSet = 2
FRM_Graficar.Graph1.GraphData = Val(linea)
x2 = Val(linea)
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 3
FRM_Graficar.Graph1.ThisSet = 3
FRM_Graficar.Graph1.GraphData = Val(linea)

FRM_Graficar.Graph1.ThisPoint = 3
FRM_Graficar.Graph1.ThisSet = 4
FRM_Graficar.Graph1.GraphData = Abs(x1 - x2)

'#4
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.LabelText = linea
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.GraphData = Val(linea)
x1 = Val(linea)
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 2
FRM_Graficar.Graph1.GraphData = Val(linea)
x2 = Val(linea)
Line Input #4, linea

```

Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 3
FRM_Graficar.Graph1.GraphData = Val(linea)

FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 4
FRM_Graficar.Graph1.GraphData = Abs(x1 - x2)

FRM_Graficar.Show 1
Loop
Close
Exit Sub
final1:
FRM_Graficar.Show 1
Exit Sub
End Sub

```

GRAFICAR RUTAS

```

Sub Graficar_Rutas ()
Dim Path As String, Fecha As String
Dim linea As String, titulo As String
Dim x1 As Single, x2 As Single
Path = App Path
flag_cancelar = False
If Right$(Path, 1) <> "\" Then Path = Path + "\"
Path = Path + "temp.txt"
Open Path For Input Shared As #4
Line Input #4, linea
Line Input #4, linea
Fecha = linea
titulo = "Trafico por Rutas Fecha " + Fecha
Load FRM_Graficar

```

```

Do While Not EOF(4) And flag_cancelar = False
  On Error GoTo final:
  FRM_Graficar.Graph1.GraphTitle = titulo
  FRM_Graficar.Graph1.DataReset = 9
  FRM_Graficar.Graph1.NumPoints = 4
  FRM_Graficar.Graph1.NumSets = 4
  FRM_Graficar.Graph1.ThisPoint = 1
  FRM_Graficar.Graph1.ThisSet = 1
  FRM_Graficar.Graph1.LegendText = "Existente"
  FRM_Graficar.Graph1.ThisPoint = 2
  FRM_Graficar.Graph1.ThisSet = 2
  FRM_Graficar.Graph1.LegendText = "Ideal"
  FRM_Graficar.Graph1.ThisPoint = 3
  FRM_Graficar.Graph1.ThisSet = 3
  FRM_Graficar.Graph1.LegendText = "Ofrecido"
  FRM_Graficar.Graph1.ThisPoint = 4
  FRM_Graficar.Graph1.ThisSet = 4
  FRM_Graficar.Graph1.LegendText = "Diferencia"

```

Line Input #4, linea

Line Input #4, linea

```
FRM_Graficar.Graph1.ThisPoint = 1
```

```
FRM_Graficar.Graph1.ThisSet = 1
```

```
FRM_Graficar.Graph1.LabelText = linea
```

Line Input #4, linea

Line Input #4, linea

```
FRM_Graficar.Graph1.ThisPoint = 1
```

```
FRM_Graficar.Graph1.ThisSet = 1
```

```
FRM_Graficar.Graph1.GraphData = Val(linea)
```

```
x1 = Val(linea)
```

Line Input #4, linea

Line Input #4, linea

```
FRM_Graficar.Graph1.ThisPoint = 1
```

```
FRM_Graficar.Graph1.ThisSet = 2
```

```
FRM_Graficar.Graph1.GraphData = Val(linea)
x2 = Val(linea)
```

```
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.ThisSet = 3
FRM_Graficar.Graph1.GraphData = Val(linea)
```

```
FRM_Graficar.Graph1.ThisPoint = 1
FRM_Graficar.Graph1.ThisSet = 4
FRM_Graficar.Graph1.GraphData = Abs(x1 - x2)
```

```
'#2
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.LabelText = linea
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.GraphData = Val(linea)
x1 = Val(linea)
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.ThisSet = 2
FRM_Graficar.Graph1.GraphData = Val(linea)
x2 = Val(linea)
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 2
FRM_Graficar.Graph1.ThisSet = 3
```



```
FRM_Graficar.Graph1.GraphData = Val(linea)
```

```
FRM_Graficar.Graph1.ThisPoint = 2
```

```
FRM_Graficar.Graph1.ThisSet = 4
```

```
FRM_Graficar.Graph1.GraphData = Abs(x1 - x2)
```

```
'#3
```

```
Line Input #4, linea
```

```
Line Input #4, linea
```

```
FRM_Graficar.Graph1.ThisPoint = 3
```

```
FRM_Graficar.Graph1.ThisSet = 1
```

```
FRM_Graficar.Graph1.LabelText = linea
```

```
Line Input #4, linea
```

```
Line Input #4, linea
```

```
FRM_Graficar.Graph1.ThisPoint = 3
```

```
FRM_Graficar.Graph1.ThisSet = 1
```

```
FRM_Graficar Graph1 GraphData = Val(linea)
```

```
x1 = Val(linea)
```

```
Line Input #4, linea
```

```
Line Input #4, linea
```

```
FRM_Graficar Graph1.ThisPoint = 3
```

```
FRM_Graficar Graph1.ThisSet = 2
```

```
FRM_Graficar Graph1.GraphData = Val(linea)
```

```
x2 = Val(linea)
```

```
Line Input #4, linea
```

```
Line Input #4, linea
```

```
FRM_Graficar Graph1 ThisPoint = 3
```

```
FRM_Graficar.Graph1.ThisSet = 3
```

```
FRM_Graficar Graph1 GraphData = Val(linea)
```

```
FRM_Graficar Graph1 ThisPoint = 3
```

```
FRM_Graficar.Graph1.ThisSet = 4
```

```
FRM_Graficar Graph1 GraphData = Abs(x1 - x2)
```

```
'#4
```

```
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.LabelText = linea
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 1
FRM_Graficar.Graph1.GraphData = Val(linea)
x1 = Val(linea)
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 2
FRM_Graficar.Graph1.GraphData = Val(linea)
x2 = Val(linea)
Line Input #4, linea
Line Input #4, linea
FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 3
FRM_Graficar.Graph1.GraphData = Val(linea)

FRM_Graficar.Graph1.ThisPoint = 4
FRM_Graficar.Graph1.ThisSet = 4
FRM_Graficar.Graph1.GraphData = Abs(x1 - x2)

FRM_Graficar.Show 1
Loop
Close
Exit Sub
final:
FRM_Graficar.Show 1
Exit Sub
End Sub
```

