

13
201



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Escuela Nacional de Estudios Profesionales
"ARAGON"

DISEÑO DE LA PROGRAMACION PARA EL PROCESADOR CENTRAL DE UNA CENTRAL TELEFONICA DIGITAL TIPO SPC

T E S I S

Que para obtener el Título de:

INGENIERO MECANICO ELECTRICISTA

Presenta:

JOSE MARIO MIRANDA MARTINEZ

TESIS CON
FALLA DE ORIGEN



ENEP
ARAGON

ARAGON

1991.



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

CONTENIDO

INTRODUCCION

CAPITULO I ANALISIS ESTRUCTURAL DE SERVICIOS EN LAS CENTRALES TELEFONICAS DIGITALES TIPO SPC

	Página
1.1 Principio de operación de una Central Telefónica Digital tipo SPC	1
1.2 Estructura tradicional de una Central Telefónica Digital tipo SPC	3
1.2.1 Parte de Conmutación	3
1.2.2 Parte de Control	8
1.2.3 Interconexión del Procesador y la Parte de Conmutación	11
1.2.4 Presentación de diferentes estructuras a nivel de Hardware y Software	13
1.3 Definición de los parámetros básicos generales	16
1.4 Principales ventajas de las Centrales Telefónicas Digitales tipo SPC	18

CAPITULO II DISEÑO DE LA PROGRAMACION PARA UNA CENTRAL TELEFONICA DIGITAL

	Página
2.1 Introducción al metodo de trabajo para el Diseño de software	25
2.2 Investigación del sistema	30
2.3 Diseño del sistema	31
2.4 Diseño del bloque	33
2.5 Unidad de software	34
2.6 Inspecciones	34
2.7 Prueba	35
2.8 Liberación del producto	37

CAPITULO III DISEÑO DEL SISTEMA

	Página
3.1 Límites de la función "LLAMADA EN ESPERA" 38
3.2 Especificaciones de funcionamiento de "LLAMADA EN ESPERA" 42
3.3 Descripción de la función "LLAMADA EN ESPERA" 50
3.4 Instrucciones de operación 62

CAPITULO IV DISEÑO DEL BLOQUE FUNCIONAL

	Página
4.1 Información acerca de la revisión del producto 67
4.2 Descripción bloque funcional SUSCAW "LLAMADA EN ESPERA" 67
4.3 Datos de Central "LLAMADA EN ESPERA" 78

CONCLUSIONES 86
---------------------	----------

GLOSARIO 89
-----------------	----------

BIBLIOGRAFIA 94
---------------------	----------

APENDICE A	ESPECIFICACIONES TECNICAS PARA CENTRALES TELEFONICAS DIGITALES DE TELEFONOS DE MEXICO "LLAMADA EN ESPERA"
-------------------	--

APENDICE B	BLOQUE FUNCIONAL "LLAMADA EN ESPERA" LISTA DE PARAMETROS, LISTA DE SEÑALES Y PROGRAMA FUENTE EN LENGUAJE DE ALTO NIVEL PLEX-C
-------------------	--

INTRODUCCION

El inicio de la historia de las telecomunicaciones podemos ubicarlo aun antes de la invención de la escritura, previo a las primitivas redes de mensajeros, señales de humo, etc..

A estos medios de comunicación les siguieron los correos con la escritura, y, posteriormente, a partir del descubrimiento de la electricidad se desarrollaron el telégrafo, el teléfono, la radio, el télex, la televisión, hasta llegar a las actuales técnicas especializadas de la telemetría, transmisión de datos, televisión por cable, etc..

Las demandas tanto de los usuarios, como de los prestadores de servicios, han traído como consecuencia un gran incremento en el uso de técnicas digitales así como también un crecimiento enorme de los volúmenes de información que se almacenan y transmiten.

Partiendo de estas demandas y de las posibilidades tecnológicas actuales, surge la conveniencia económica y la posibilidad técnica de crear un nuevo equipo que tenga por un lado gran capacidad de transporte y por otro, sea lo suficientemente flexible en el manejo de la información, además de poder evolucionar a partir de las redes existentes y ser capaz de adaptarse a la incorporación de servicios futuros.

Se ha pensado en la técnica digital debido a las ventajas que esta ofrece sobre la analógica, entre ellas cabe mencionar que se facilita la integración en todos los niveles de la red, ya que todas las señales son reducidas a un elemento común (en este caso el bit) sin distinción del tipo de servicio.

Con esto, resulta más factible emplear repetidores y regeneradores que introducen niveles mínimo de degradación de las señales mejorando así la calidad de los servicios.

Las aplicaciones empresariales de las centrales telefónicas digitales pueden enfocarse desde dos puntos de vista básicos:

- a) La pequeña empresa
- b) La gran empresa

INTRODUCCION

El inicio de la historia de las telecomunicaciones podemos ubicarlo aún antes de la invención de la escritura, previo a las primitivas redes de mensajeros, señales de humo, etc..

A estos medios de comunicación les siguieron los correos con la escritura, y, posteriormente, a partir del descubrimiento de la electricidad se desarrollaron el telégrafo, el teléfono, la radio, el télex, la televisión, hasta llegar a las actuales técnicas especializadas de la telemetría, transmisión de datos, televisión por cable, etc..

Las demandas tanto de los usuarios, como de los prestadores de servicios, han traído como consecuencia un gran incremento en el uso de técnicas digitales así como también un crecimiento enorme de los volúmenes de información que se almacenan y transmiten.

Partiendo de estas demandas y de las posibilidades tecnológicas actuales, surge la conveniencia económica y la posibilidad técnica de crear un nuevo equipo que tenga por un lado gran capacidad de transporte y por otro, sea lo suficientemente flexible en el manejo de la información, además de poder evolucionar a partir de las redes existentes y ser capaz de adaptarse a la incorporación de servicios futuros.

Se ha pensado en la técnica digital debido a las ventajas que ésta ofrece sobre la analógica, entre ellas cabe mencionar que se facilita la integración en todos los niveles de la red, ya que todas las señales son reducidas a un elemento común (en este caso el bit) sin distinción del tipo de servicio.

Con esto, resulta más factible emplear repetidores y regeneradores que introducen niveles mínimo de degradación de las señales mejorando así la calidad de los servicios.

Las aplicaciones empresariales de las centrales telefónicas digitales pueden enfocarse desde dos puntos de vista básicos:

- a) La pequeña empresa
- b) La gran empresa

Si agrupamos dentro de las pequeñas empresas a las pequeñas fábricas, oficinas, agencias de viajes, centros de servicios, gasolineras, etc., encontraremos en este ambiente algunos teléfonos digitales y, frecuentemente, una computadora personal, lo cual da la posibilidad de alojar software para teletex, correo electrónico o telefax para velocidad de hasta 2.048 Mb/seg.

Así, las aplicaciones básicas en este contexto son:

- Telefonía básica, complementada con funciones de texto en pantalla digital permitiendo mostrar información tal como "libre" u "ocupado", el número marcado, el cargo por la llamada, etc..
- Intercambio de correspondencia de negocios (correo electrónico) con otras empresas.
- Intercambio de planos, documentos o dibujos con otras empresas.
- Bases de datos comunes para búsqueda de información.

En las grandes corporaciones cuya característica general es la dispersión geográfica de oficinas y fábricas como son los bancos, hospitales, instituciones públicas y privadas, compañías de seguros, grandes almacenes, etc., se cuenta con un gran número de teléfonos digitales, PC's, terminales de télex, teletex, facsimil, etc..

Las aplicaciones son en general las mismas que para la pequeña empresa, pero con una complejidad mucho mayor y una demanda muy superior de capacidad de transferencia, por ejemplo, en la comunicación entre computadoras, videoconferencias, etc..

Los beneficios que se han obtenido en base a experiencias con grandes empresas son:

- Tiempos de respuesta menores.
- Reducciones significativas en el número de controladores de terminales.
- Terminales más sencillas y baratas para el acceso a las diferentes redes.
- Alta calidad en la red digital de voz.
- Teléfonos de funciones múltiples.
- El uso de pares trenzados para voz y datos.

- Conexión con otros conmutadores PBX con múltiples funciones.
- Reducción en los costos de conmutación.
- Conferencias telefónicas de alta calidad.
- Videoconferencias conmutadas usando videocompresión.
- Transferencia de archivos a alta velocidad.
- Integración de LAN's privadas con RDSI.

Situación Actual Mundial

En la actualidad existen ya una gran cantidad de países que están llevando a cabo pruebas de campo sobre RDSI. Estas pruebas involucran equipos de los principales proveedores de centrales digitales en el mundo.

Además se ha observado claramente una marcada tendencia de que los servicios RDSI, existen de manera comercial y en forma generalizada a partir de 1990

Situación en Teléfonos de México

Como un resultado de las investigaciones sobre las estrategias y experiencias que se están desarrollando en el mundo para evolucionar hacia RDSI, se diseñó una estrategia en TELMEX, la cual, se divide en las siguientes partes:

1.- Aumento de la capacidad de la red digitalizándola sincronizándola y utilizando señalización por canal común, la cual, le dará a la red un sistema rápido y eficiente que permitirá manejar facilidades que no se puedan dar actualmente con los sistemas de señalización tradicionales

2.- Diversificación de Servicios, la RDSI, implica una integración de servicios sin embargo antes de integrar es necesario contar con los servicios e identificar a aquellos que son importantes y que serán parte medular de esta red.

Estos servicios pueden ser agrupados en cuatro categorías :

- Teleservicios.
- Servicios telefónicos suplementarios.
- Red de datos.
- Red Superpuesta (Servicios sofisticados para transmitir voz y datos).

Los planes de prueba e introducción se pueden dividir en 2 partes:

Bienio 1990-91 - Demostración de facilidades RDSI a nivel nacional realizando la prueba interurbana de las facilidades RDSI.

Bienio 1992-93 - Comercializar los servicios en las ciudades donde se ofrezca Red Superpuesta (2 ciudades principales) y negociación de conexión con otras redes.

La RDSI, permitirá optimizar el uso de los sistemas digitales generalizándolos y permitiendo además integrar sistemas de información y acceso a bases de datos comunes en empresas cuyas oficinas estén distribuidas geográficamente.

La perspectiva de comercialización en México está planteada para el bienio 1992-1993, sin embargo, el plan podría adelantarse uno o dos años si los resultados de las pruebas son satisfactorias.

Debido a la importancia estratégica que actualmente tienen las telecomunicaciones como soporte de los sistemas de información en México, es evidente, la necesidad de adecuar los sistemas de comunicación de tal forma que además de contar internamente con una eficiente administración de sus sistemas de información, cuente también con la infraestructura necesaria para integrarse a la RDSI a nivel nacional e internacional.

Tomando en cuenta que los actuales sistemas de telecomunicación son controlados por un computador ó procesador, es de vital importancia hacer notar que estos sistemas, dependen casi en su totalidad del buen funcionamiento de este procesador central, ya que por ejemplo, una pequeña falla en el procesador se vería

reflejada en una falla del sistema, generando perdidas incalculables.

El diseño de la programación de este procesador (diseño de software del procesador central) es la parte medular dentro del proceso de desarrollo de todo el sistema, y es precisamente en él donde la calidad y confiabilidad del mismo están basadas.

El presente trabajo de tesis consiste en el diseño de software para el procesador central de una central telefónica y, su principal objetivo es demostrar que no obstante que el diseño de software es un proceso muy delicado, sofisticado y de alto nivel tecnológico, se tiene la capacidad para desarrollarlo y obtener resultados de gran calidad y confiabilidad, a la altura de cualquiera en el mundo.

Para ello se ha dividido dos partes :

- I - En la primera se ofrece al lector una pequeña introducción a las centrales telefónicas digitales tipo SPC, al establecimiento de llamadas en este tipo de centrales, a la importancia del software del procesador central así como también algunos modelos de diseño de software.
- II.- La segunda parte consta del diseño del software de la facilidad de usuario LLAMADA EN ESPERA basada en el método RUDOLPH, el cual comprende, desde el estudio de los requerimientos hasta la elaboración de la unidad de software, la cual está escrita en lenguaje de alto nivel (PLEX).

Cabe señalar que la información aquí contenida es lo más actual y veraz que se pudo obtener, ya que por ejemplo, en dependencias oficiales la información estadística que se maneja difiere de lo obtenido en otras, y la información que se maneja en las dependencias privadas es actual pero muy difícil de obtener debido a que es manejada con carácter de confidencial.

CAPITULO I

ANALISIS ESTRUCTURAL DE SERVICIOS EN LAS CENTRALES TELEFONICAS DIGITALES TIPO SPC

1.1 PRINCIPIO DE OPERACION DE UNA CENTRAL TELEFONICA DIGITAL TIPO SPC

Un sistema SPC (Stored Program Control) ó Sistema de Control por Programa almacenado, es aquel sistema en el cual todas sus operaciones son controladas por un procesador, lo que significa, que programas almacenados en la memoria de un computador controlan la operación del sistema.

Una Central Telefónica Digital tipo SPC es aquella central telefonica en la que todas sus operaciones son controladas por un procesador, el cual, es un computador trabajando en tiempo real y que controla el proceso telefónico.

La introducción de esta técnica de control por programa almacenado a la ingeniería de centrales telefónicas inició una nueva era en el campo de las telecomunicaciones, debido a la gran versatilidad que adquiere la central telefónica al poder ofrecer una gran confiabilidad y rapidez de operación así como también una gran cantidad de facilidades al usuario, al operador y al administrador de la misma.

Una característica de la técnica SPC, en telefonía, comparada con la técnica de control convencional, es que las señales de entrada externas no inician por si mismas ninguna actividad en el sistema de control de una central SPC.

El sistema de control debe explorar regularmente todas las líneas y órganos que reciban señales externas (líneas de abonado, líneas de enlace, transmisores de código, dispositivos de enlace, unidades entrada/salida) de tal forma que no se pierda la detección de ninguna señal de llegada.

Este trabajo es por si mismo rutinario y para realizarlo no se requiere de un alto grado de inteligencia, pero a la vez, representa una gran parte del trabajo total, por lo tanto, en algunos sistemas SPC, este trabajo ha sido asignado a exploradores especiales, reduciendo así considerablemente la carga del procesador central dando como resultado que éste se encargue de

tareas más complejas, logrando así una utilización más eficiente control central.

Este explorador debe diseñarse de tal manera que tenga su propio almacén con las imágenes de los puntos de prueba ó exploración, y las direcciones de las líneas ó órganos explorados, en las que ha detectado las señales.

La exploración de un sistema se puede dividir dependiendo del tipo de procesamiento requerido, de acuerdo a dos principios fundamentales, llamados PRINCIPIO DE DIVISION DE FUNCIONES Y PRINCIPIO DE SEGMENTACION de donde :

- PRINCIPIO DE DIVISION DE FUNCIONES : Aquí se utilizan varios procesadores, y cada uno de estos procesadores atiende a un limitado número de funciones pero, para toda la central telefónica.

PRINCIPIO DE SEGMENTACION : Aquí se utilizan varios procesadores, pero, en este caso el procesador ejecuta todas las funciones para una sola parte de la central telefónica, adaptable en forma de módulos de extensión.

La ventaja de la división por funciones es que cada uno de los procesadores puede ser bien adaptado para llevar a cabo las funciones particulares asignadas, la desventaja es que desde el principio el procesador debe de estar dimensionado para la capacidad final de la central, limitando de esta forma el crecimiento de la central telefónica; un ejemplo de esta aplicación es el SISTEMA 12 desarrollado por la empresa INDETEL.

La división por módulos de extensión demanda menos capacidad de cada procesador, pero en este caso el procesador debe estar con el almacén de programas completo, y la interconexión entre procesadores para el caso en que la conexión pase a través de varios módulos de extensión, es decir involucra carga extra; una aplicación de este sistema se puede observar en el sistema AXE 10, desarrollado por la empresa ERICSSON.

1.2 ESTRUCTURA TRADICIONAL DE UNA CENTRAL TELEFONICA DIGITAL SPC

Básicamente una central telefónica digital tipo SPC esta dividida en dos grandes areas:

1- DIVISION CIRCUITERIA (hardware) la cual desarrolla sus funciones basada en la técnica de circuitos y con técnica de procesamiento de datos.

2- DIVISION FUNCIONAL basada en las reglas de comportamiento de la central.

Una estructura particular sería la siguiente :

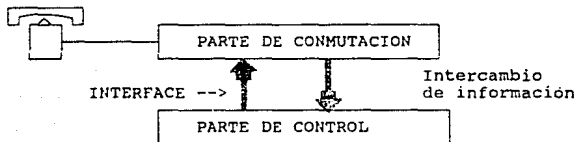


Fig 1.1 ESTRUCTURA TRADICIONAL DE UNA CENTRAL TELEFONICA DIGITAL SPC

1.2.1 PARTE DE CONMUTACION

Es la parte de la central telefónica donde las líneas de abonados se interconectan. Contiene los elementos sobre los cuales tiene lugar la conexión de habla y la transmisión de señales (red de conmutación), así como también los circuitos para funciones simples de telefonía, tales como generadores de tonos, circuitos para la recepción de señales de monitoreo de cambio de estado y la traducción de estas señales a una forma adecuada para la parte de control.

A continuación en la figura 1.2 se presenta el diagrama funcional de un microteléfono, ya que por medio de este dispositivo la secuencia de conmutación es iniciada.

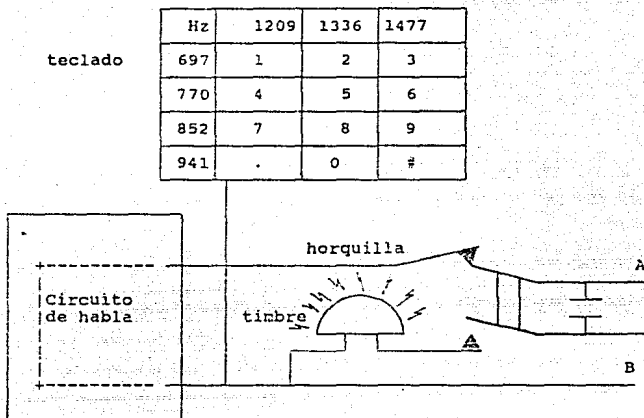


Fig. 1.2 Diagrama funcional de un microteléfono

El teléfono está conectado via los relevadores de conmutación a un número de dispositivos con diferentes funciones, por ejemplo, para una conversación a el dispositivo ABJ (AB junctor ó connector de A a B), para enviar señales desde su teclado a un KR (key set receiver, receptor de teclado) ó a un BS (busy sender, transmisor de tono de ocupado) si la conversación no llegara a establecerse.

A continuación se describen brevemente algunos de los dispositivos más importantes que se usan durante la secuencia de conmutación como son :

La interface para la línea de abonado (LI), el dispositivo ABJ por medio del cual se lleva a cabo la conversación y el dispositivo para la recepción de los dígitos (KR).

DISPOSITIVO LI

Se requiere para enviar señales de bucle de los abonados, todos estos dispositivos y los selectores de relevadores están controlados por el procesador via el bus de datos, este dispositivo LI contiene compuertas mediante las cuales el procesador se informará acerca de los estados de los contactos de horquilla de los abonados, el chequeo tiene lugar con ordenes enviadas desde el procesador hasta la entrada de control de las compuertas.

Las compuertas son de tipo especial, en las cuales los estados lógicos de las entradas conectadas a las líneas de abonado están caracterizadas por voltajes diferentes a los normales (0V +5V), esto es debido a que la alimentación de abonado requiere de voltajes mayores a los utilizados en circuitos lógicos.

CONECTOR ABJ

En algunos sistemas se le conoce como SNR (juego de relevadores del circuito de cordón) tiene como principal función la de juntar las corrientes de voz, pero, separadas de sus respectivos corrientes de D.C.; en esta forma el estado del contacto de horquilla de uno de los abonados no afectara la lectura del estado de la horquilla del otro abonado, el conector ABJ atiende también la alimentación microfónica de ambos abonados A y B, además de contener los circuitos para el envío de señales de llamada para el abonado que recibe la llamada (abonado B) y tono de llamada para el que la realiza (abonado A), circuitos de corte de estos tonos y señales una vez que el abonado B conteste.

CIRCUITO KR

Contiene filtros para las siete frecuencias del teclado y un generador de frecuencia para el tono de marcar, cuando los generadores de frecuencia en el aparato telefónico del abonado están conectados a los hilos "a" y "b" los filtros que corresponden a estas frecuencias darán un 1 lógico, las salidas de los filtros se conectan a compuertas NAND, cuyas salidas darán el número recibido en forma de un 0 ó en 1, en uno de los 8 hilos conectados al bus de datos, de esta forma la central puede recibir los números de los diferentes abonados con lógica cableada para hacer más sencillo el entendimiento de los dígitos por medio del procesador, generalmente se usa la técnica de datos para la decodificación, conectándose el bus del procesador directamente a las salida de los filtros, via una compuerta por filtro.

Una vez que A ha recibido tono de marcar y ha marcado el número de B, el tono de marcar tendrá que suspenderse, esto se logra desconectando el tono de marcar, inmediatamente que el subscriptor A ha marcado el primer dígito, lo cual requiere de un punto de operación en cada uno de los KRs dados para este propósito.

SECUENCIA DE CONMUTACION EN LA CENTRAL TELEFONICA DIGITAL BPC

Cuando un abonado hace una llamada (A descuelga su microteléfono), es detectado por el procesador el cual a intervalos regulares (cada 10 milisegundos) lee el estado de los abonados via el LI.

El procesador interpreta el cambio de un bucle abierto a un bucle cerrado como un intento de llamada. Este debe, por lo tanto, recordar el resultado de la prueba anterior.

Cuando el procesador ha detectado el intento de llamada conecta al abonado a un KR, éste le mandará tono de marcar, entonces el abonado A marcará el número de abonado B el cual es identificado por medio de filtros y leído por el procesador, por medio del conector ABJ.

AB son conectados y por medio de ABJ el procesador inicia la señal de timbre hacia el abonado B hasta que éste responde.

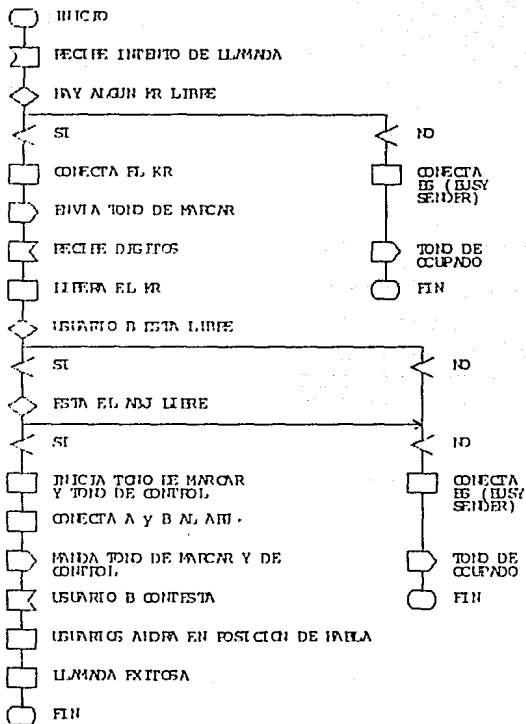
Cuando la conversación termina los abonados reponen sus microteléfonos y el procesador libera la conexión.

El procesador detecta si hay algún trabajo a realizar leyendo en LI y KR, via el LI el procesador puede detectar llamadas, obtener respuesta del microteléfono y repuesta de B, via el KR, el procesador puede detectar los dígitos que llegan a la central.

Si el KR, el ABJ ó el subscriptor B están ocupados con otro abonado entonces el abonado A es conectado a un BS que le dara tono de ocupado.

A continuación veremos con un poco más de detalle la secuencia de conmutación

1 SECUENCIA_DE_UNA_LLAMADA



1.2.2 LA PARTE DE CONTROL

Es la parte donde se encuentran los circuitos y programas que atienden las funciones más complejas (inteligentes) de la central, tales como identificación e interpretación de los cambios de estado de la parte de conmutación, de acuerdo, a los programas que están basados en los requerimientos del comportamiento de la central en diferentes situaciones.

La parte de control de una central telefónica digital esta constituida por el procesador central y éste a su vez esta compuesto de 3 partes principales :

- 1.- Unidad central de procesamiento ó CPU (Central Processing Unit)
- 2.- Almacen de programas ó PRS (Program Store)
- 3.- Almacen de Datos ó DAS (Data Store)

La interconexión de estas tres partes es como sigue :

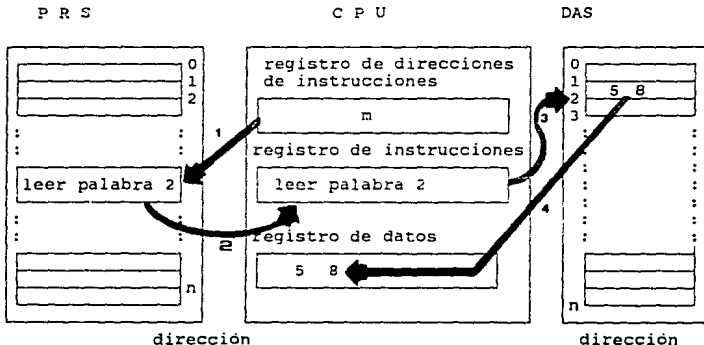


fig. 1.3 PARTE DE CONTROL DE UNA CENTRAL SPC

Su principio de operación es el siguiente :

El CPU contiene un registro llamado Registro de Dirección de Instrucciones (Instruction Address Register), el cual, contiene la dirección para una palabra (instrucción) en el almacén de programas. Cuando esta dirección se envía al PRS, el CPU obtiene como respuesta el contenido de la dirección indicada, es decir recibe una instrucción por ejemplo, para la dirección m encontraremos que la instrucción ahí almacenada es "leer la palabra 2", la instrucción no está en "texto claro", sino que en código binario.

La instrucción leída será situada en el registro de instrucciones. El CPU interpreta el código binario y toma cuenta que se debe realizar una lectura en dirección 2 del almacén de datos. La dirección será enviada al almacén de datos.

Como respuesta obtendremos el contenido de esta dirección en el registro de datos, quedando así completa la ejecución de la instrucción.

La palabra que ha sido leída puede ser procesada por la siguiente instrucción, en general las instrucciones son ejecutadas en el mismo orden en que se encuentran situadas en el almacén de programas, y de esta manera el CPU necesita únicamente adelantar un paso el contenido de su registro de direcciones e instrucciones para alcanzar su siguiente instrucción.

PROCESAMIENTO DE INSTRUCCIONES EN PROCESADOR CENTRAL

Las instrucciones en PRS están dadas en forma de palabras de código binario, la instrucción por ejemplo "READ WORD 2" (leer la palabra 2) puede dividirse en dos partes, el código de Operación y parte Paramétrica.

El código de operación indica el TIPO DE OPERACION, en este caso sería LEER, mientras, que la parte Paramétrica indica DONDE SE VA A REALIZAR LA OPERACION.

Cuando se escribe un programa con un gran número de instrucciones, se prefiere un método corto y conciso de escritura.

Por esta razón a cada instrucción se le da una sigla por ejemplo Read Word from Store (leer una palabra del almacén) sería RWS y el parámetro sería cualquier número por ejemplo X, este método de escritura se llama código base (SOURCE CODE) o también ensamblador (ASSEMBLER), antes de escribir esta instrucción en un programa, debe ser convertida primero a código binario, a esta

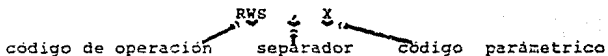
conversión se le llama ensamblar, y el código binario se llama código de máquina (OBJET CODE ó MACHINE CODE).

A continuación se muestra la secuencia que se realiza el CPU para procesar una instrucción.

FUNCIÓN : En DAS la palabra en dirección X será escrita en el registro de datos en CPU

Sigla Read Word From Store

Código ensamblador : como modelo

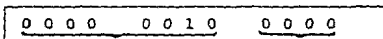


-como se escribe en el programa, en el cual deseamos leer una palabra con dirección 2

RWS,2

Código de máquina (código binario) ,asumimos que la instrucción

RWS tiene el código de operación No. 0



código paramétrico = 2

código de operación = 0

Fig. 1.4 INSTRUCCIONES EN EL PROCESADOR CENTRAL

Con frecuencia se leen ó escriben varias posiciones de bit en paralelo, las que luego constituyen una PALABRA, la cual consta de puntos de prueba llamados Palabra de Prueba y también consta de puntos de operación llamados Palabra de Operación, las señales requeridas para la lectura ó escritura de información constan de bits de direcciones y de bits de operación.

La dirección de palabra indica la palabra a ser leída ó escrita, y la dirección de operación (orden de lectura REO u orden de escritura WRO), dá la función a realizar, es decir si se va a leer ó escribir. Cada dirección una

palabra de una determinada cantidad de bits, por ejemplo 8, la que será escrita ó leída.

1.2.3 INTERCONEXION DEL PROCESADOR Y LA PARTE DE CONMUTACION

La parte de conmutación generalmente consta de circuitos electromecánicos (relevadores) y circuitos lógicos electrónicos, mientras que la parte de control utiliza la técnica electrónica de procesamiento de datos con programas y datos almacenados, debido a esta diferencia de componentes, los tiempos de operación son bastante grandes es decir milisegundos en la parte de conmutación y microsegundos en la de control, lo cual hace necesario el uso de una interface a base de circuitos electrónicos capaces de reaccionar a nivel de microsegundos y que posea un elemento de memoria capaz de retener la orden recibida por la orden de control y la transfiera a la parte de conmutación que es más lenta, como lo vemos en la siguiente figura :

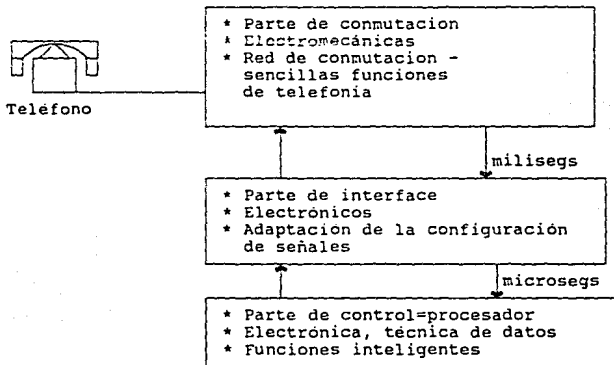


Fig. 1.4 TIEMPOS DE OPERACION EN UNA CENTRAL TELEFONICA

En esta parte el procesador verifica, via la interface, lo que sucede en la parte de conmutación, usando señales de dirección para abrir una compuerta obtendremos el estado del relevador, a esto se le llama Prueba de Lectura y la respuesta conseguida de esta prueba es el Dato de Lectura, este principio muestra que una señal de

la parte de conmutación no afecta automáticamente al procesador, por que la señal no se detecta si no hasta que el procesador se "decide" a leerla, esto es; el procesador realiza lecturas en puntos de prueba y a intervalos regulares.

La interface necesita un elemento de memoria que opere los relevadores (por ejemplo un "flip-flop"), de tal manera, que cuando el procesador ordena alguna operación, indicando una dirección, el "flip-flop" controle al relevador indicado, el "flip-flop" será fijado ó despejado con el dato de escritura en el en forma de 0 ó 1, despues de ésta orden, el "flip-flop" retiene su estado y opera el relevador.

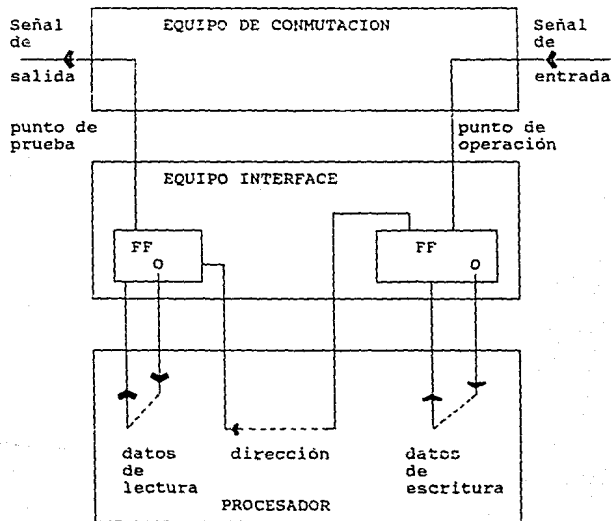


Fig. 1.6 INTERFACE PROCESADOR-CONMUTADOR

Esta interface consta principalmente de un registro de direcciones (DEA) y decodificadores de direcciones (COD)

así como varios circuitos de compuertas y flip-flops en los puntos de cruce, cabe mencionar que actualmente algunos dispositivos telefónicos contienen también compuertas que desde el punto de vista funcional, pertenecen a la interface, pero que por razones prácticas, se incluyen en los dispositivos que pertenecen a la parte de conmutación.

Así tenemos que el direccionamiento lo inicia el procesador enviando la dirección de órgano hasta el registro DEA ó registro de direcciones (device Address register), esta dirección se decodifica en uno de los circuitos COD, uno de ellos se activa con la orden de LECTURA y otro con la de ESCRITURA, las salidas de estos activan las compuertas de entrada de los diferentes dispositivos telefónicos, por medio de los cuales, los datos se llevan dentro ó fuera de los dispositivos via el bus de datos.

1.2.4 PRESENTACION DE DIFERENTES ESTRUCTURAS A NIVEL DE HARDWARE Y SOFTWARE

A continuación se muestran dos ejemplos referentes al establecimiento de una llamada telefónica normal, en centrales SPC, que actualmente se ofrecen en el mercado nacional.

OPERACION DEL SISTEMA AXE-10

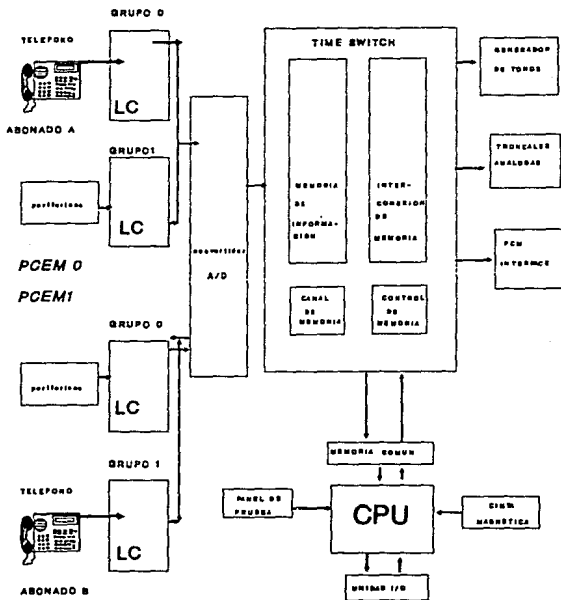
- 1.- Abonado levanta la bocina de su teléfono.
- 2.- LIC detecta intento de llamada y por medio de LI2, JT/RT y CJ reserva un canal para la conversación
- 3.- EMTS por medio de TS, KR2 y CJ verifica si el subcriptor tiene asignadas algunas facilidades.
- 4.- KRC usa KR2, CJ, SC y RE también manda el tono de invitación a marcar y para recibir, almacenar los digitos marcados.
- 5.- DA y RA anlizan los digitos y las posibles rutas hacia el abonado solicitado, CA inicia la tasación.
- 6.- BT Selecciona la ruta, a través de un canal común CCS7.
- 7.- GS a través de ST7 y ETC logra comunicación con el abonado destino.

OPERACION DEL SISTEMA OMNI-GTD-1000

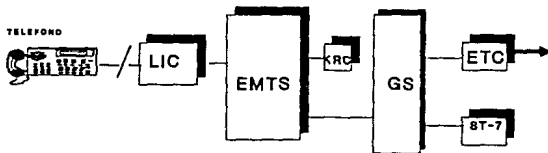
- 1.- Abonado levanta la bocina de su teléfono.
- 2.- Line circuit detecta intento de llamada
- 3.- A través de Time Switch almacena e identifica la identidad del usuario.
- 4.- CPU controla el sistema distribuyendo cargas de trabajo, cantidad de memoria y monitoreando el sistema.
- 5.- Digital tone source provee los tonos solicitados por el CPU.
- 6.- PCEM realiza la señalización por canal común.

GTE OMNI-1000

DIAGRAMA A BLOQUES



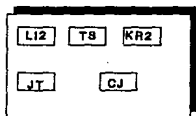
CENTRAL TELEFONICA DIGITAL SPC TIPO AXE10
BLOQUES QUE INTERVIENEN EN UNA LLAMADA



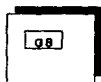
HARDWARE

SOFTWARE

SSS



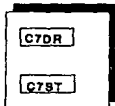
GSS



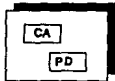
TSS



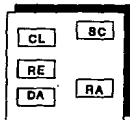
CCS



CHS



TCS



1.3 DEFINICION DE LOS PARAMETROS BASICOS GENERALES**SISTEMAS DE TRANSMISION (TDM, PCM, FDM)**

Básicamente existen 3 técnicas para transmisión en telefonía :

- 1) Pares Físicos
- 2) Multiplexión por División de Frecuencia (FDM)
- 3) Multiplexión por División de Tiempo (PCM)

1) PARES FÍSICOS

Este sistema llamado pares físicos es como su nombre lo indica, es un sistema de transmisión voz por medio de un par físico de cables, que fue el sistema que se empleó en los principios de la telefonía, y aún se sigue empleando en pequeñas localidades rurales donde la distancia hacia la central telefónica no es mayor de 10 km., pero obviamente éste tiende a desaparecer en un futuro muy cercano.

La disminución de la atenuación de una línea se logra aumentando su inductancia, y, se logra intercalando cada 2000 metros una bobina de denominadas Bobinas Pupin debido a que quien puso en práctica por primera vez esta técnica fue el norteamericano de origen húngaro Pupin.

2) MULTIPLEXION POR DIVISION DE FRECUENCIA (FDM)

Al crecer la cantidad de enlaces telefónicos, las líneas no tardaron en encarecerse y hacerse inmanejables, esto dio pie al rápido desarrollo del método denominado Multiplexaje por División de Frecuencia (FDM Frequency División Multiplex) ó también conocido como sistema de transmisión de frecuencias portadoras,

La idea de modulación es convertir la señal de frecuencia de voz en alguna otra señal eléctrica más fácil ó más barata de transmitir.

Para obtener una señal modulada se requieren dos señales, la señal A que corresponde a la señal del habla (la que modula) y la señal B que corresponde a una señal portadora, que consiste de una sola frecuencia, las señales portadoras son siempre de una frecuencia más alta, la señal C, es la señal modulada.

De ésta manera es como por una línea telefónica se pueden enviar, miles de conversaciones, cada una modulada por un frecuencia portadora diferente.

3) MULTIPLEXION POR DIVISION DE TIEMPO (PCM)

El sistema de Multiplexión por división de tiempo (PCM Pulse Code Modulation), es empleado para aumentar la cantidad de enlaces en los cables ya existentes entre centrales, a intervalos pequeñísimos se realizan muestreos de cada una de las conversaciones, mismos que se envían uno tras otro, al lado receptor como si fuesen rompecabezas.

Un sistema telefónico consta de 32 canales que se transmiten físicamente por el mismo par físico, para evitar la pérdida de información en las conversaciones se obtienen 8000 muestras por segundo, y el tiempo requerido para obtener el muestreo de la siguiente señal es de 3.9 microsegundos.

De lo anterior tenemos que si son 32 canales con un tiempo de muestreo de 3.9 microsegundos, un cuadro ó tren de información para ser enviado al lado receptor toma 125 microsegundos.

De los 32 canales, 30 se emplean para conversación ya que el primer cuadro se emplea para indicar el inicio del mismo y el 16avo cuadro se emplea para señalización.

Los muestreos de las señales reales ó analógicas, de cada una de las conversaciones telefónicas se envían ya transformadas en trenes de 8 caracteres binarios, y debido a ésta característica, la señal es mucho menos sensible al ruido, diafonía y distorsión.

La siguiente gráfica nos dá la comparación de los 3 sistemas de transmisión en relación costo-distancia.

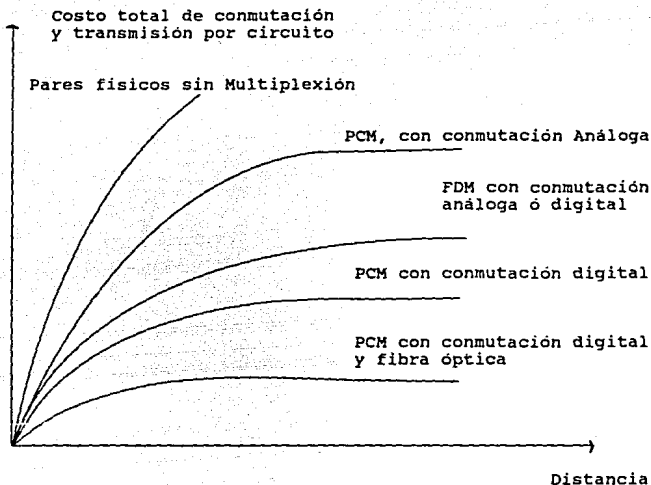


Fig. 1.6 Curvas Costo-distancia para diferentes medios de transmisión

1.4 PRINCIPALES VENTAJAS DE LAS CENTRALES TELEFONICAS DIGITALES TIPO SPC

CONFIABILIDAD

La confiabilidad de una central telefónica SPC se basa en los siguientes puntos:

- Baja tasa de fallas mediante el uso de componentes y soluciones al sistema , bien ensayados (MTBF del equipo).
- Duplicación de los órganos centrales más importantes (Equipo redundante ó tipo 1+1)
- Rapidez para detección las fallas mediante una supervisión más eficiente (monitoreo desde terminales)

-Eliminación de fallas mediante indicación automática de las mismas en cuanto sea posible, (mantenimiento automatizado).

MÉTODOS DE DETECCIÓN DE FALLAS EN UN SISTEMA SPC

- Control de paridad de los datos transportados y almacenados.
- Supervisión de tiempo en las secuencias
- Chequeo de plausibilidad de datos y programas
- Rutinas de prueba logradas mediante programas especiales
- Comparación continua entre órganos duplicados que trabajan en forma paralela-sincrónica
- Supervisión del tráfico real ó del tráfico de prueba generado.

EL DISEÑO DE SOFTWARE O PROGRAMACION DE LA CENTRAL SPC

El procesador Central de una central telefónica SPC es programado por medio de lenguajes de alto nivel (PASCAL, PLEX, LENGUAJE C) ó por medio de programas ensambladores cuando se requiere tener un control más preciso de las direcciones absolutas de cada instrucción (ASA, microinstrucciones).

Este sistema de programación no es solamente hacer programas, si no, que involucra todo un largo proceso de diseño, el cual implica desde el estudio de los requerimientos internos y externos del sistema hasta la prueba e instalación del producto.

Así como también la forma de expresar en documentos todos los datos que están cargados en los almacenes de la central telefónica.

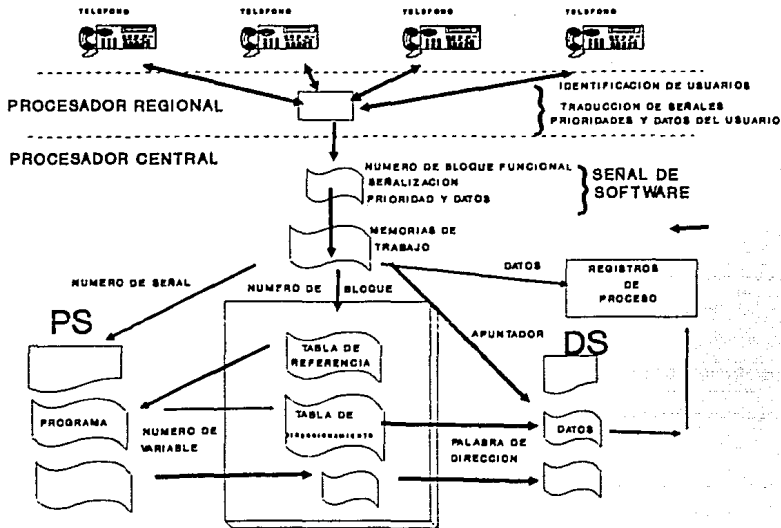
Después de introducir toda la información básica para una central, se arranca el proceso de generación, en el cual las instrucciones de alto nivel son traducidas a código de máquina por medio de un compilador.

Posteriormente se carga la central, es decir, se activa primero el monitor de carga, que es el que localiza los programas y construye las tablas de referencia del procesador central, llena dichas tablas con direcciones, posteriormente se introduce a la central los datos por medio de un sistema de entrada/salida con lo cual tenemos la central lista para entrar a operación.

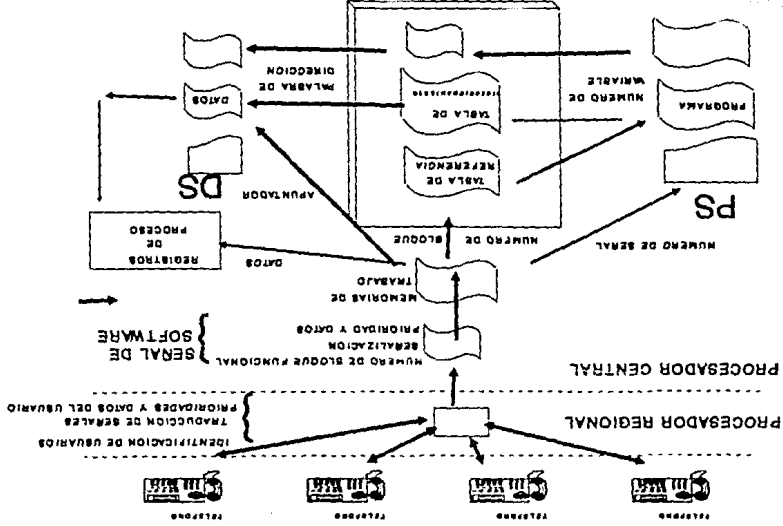
La figura de la página siguiente muestra de manera general la interrelación entre los dispositivos telefónicos, el procesador central y su software.

RELACION USUARIO-PROCESADOR-SOFTWARE

ANALISIS ESTRUCTURAL



RELACION USUARIO-PROCESADOR-SOFTWARE



SERVICIOS QUE ACTUALMENTE SE OPRECEN AL USUARIO

Basándose en las necesidades de cada grupo podemos clasificarlos básicamente en 3 grupos:

- a) Los administradores de la central telefónica
- b) los operadores de la central telefónica
- c) Los usuarios de la central telefónica (abonados)

Cada uno de estos usuarios tiene necesidades telefónicas diferentes y obviamente intereses diferentes, por ejemplo :

-A un administrador le interesa que su cuando su equipo sufre una falla, ésta, sea fácil de localizar y de reparar ó que esto se haga automáticamente.

-A un operador le interesaría enrutar tráfico de la central hacia otra central cuando esta muy congestionada y así poder brindar un servicio más eficiente.

-A un usuario normal le interesa que si ésta hablando por teléfono y otra persona le ésta llamando al mismo tiempo, por medio de un tono él pueda saber que alguien le está llamando y evitar perderse esa llamada.

A continuación listamos algunos de los más importantes servicios que la central telefónica digital SPC brinda a sus usuarios.

a) Los administradores de la central telefónica

- Larga vida de duración del equipo
- Renovación por etapas
- Flexible, que se pueda expandir de acuerdo al incremento de usuarios
- Simple de mantener y reparar
- Esquemas de numeración
- Distribución automática del tráfico
- Información de las clases de servicios
- Reportes de líneas con problemas ó defectuosas
- Tasación
- Reportes de tasación
- Alarmas de congestiónamiento
- Centralización de mantenimiento
- Transferencia automática de datos cuando existe una falla de energía
- Control del número marcado
- Rastreo de llamadas maliciosas

b) los operadores de la central telefónica

- Establecer una llamada
- Contestar una llamada
- Transferir una llamada
- Retener una llamada
- Música durante la retención de una llamada
- Establecer una conferencia múltiple
- Modificar fechas y tiempos en la central
- Hacer reservación de líneas troncales
- Control de las líneas troncales
- Intercomunicación con los diferentes grupos
- Restricción de servicios a ciertos abonados.
- Liberación forzada de líneas troncales
- Acceso directo a líneas troncales
- Interrumpir un servicio como por ejemplo "no molestar"
- Asignar nuevos servicios a un usuario
- Remover estos servicios a un usuario
- Cambiar los servicios a un usuario
- Equipo de dictado

c) Los usuarios (abonados) de la central telefónica

- Bajo costo de llamadas
- Rápido establecimiento de llamadas
- Fácil acceso
- Buena calidad de transmisión
- Establecer una llamada
- Contestar una llamada
- Transferir una llamada
- Retener una llamada
- Marcación abreviada
- Despertador
- Consultas mientras esta en una conversación
- Establecer una llamada con 3 subscriptores al mismo tiempo
- Establecer una conferencia hasta con 8 subscriptores al mismo tiempo
- Rastreo de llamadas maliciosas
- No molestar
- Protección de voz y datos durante las transmisión
- Contestador automático
- Contestador nocturno
- Repetición automática del último número marcado
- Asignación de números vacantes secundarios a un subscriptor primario
- Servicio de alertamiento distintivo
- Programación de una retollamada
- Contestación de una llamada desde otro teléfono
- Sigueme
- Llamada en espera

La lista de servicios suplementarios que actualmente está identificado según la recomendación I.250 del libro azul de CCITT en su tomo III y fascículo III-7 es como sigue.

Servicios suplementarios de identificación de número.

Marcación directa de extensiones.(MDE)
 Números Múltiples de abonado.(NMA)
 Presentación de identificación de la línea llamante.(PILLN)
 Restricción de identificación de la línea llamante.(RILLN)
 Presentación de identificación de la línea conectada.(PILC)
 Restricción de identificación de la línea conectada.(RILC)
 Identificación de llamadas maliciosas.(ILLM)
 Subdireccionamiento.(SUBD)

Servicios suplementarios de Ofrecimiento de llamadas.

Transferencia de llamadas.(TLL)
 Reenvío de llamada en caso de ocupado.(RLLO)
 Reenvío de llamada en caso de ausencia de respuesta.(RLAR)
 Reenvío de llamada incondicional.(RLLI)
 Desviación de llamadas.(DLL)
 Búsqueda de línea.(BL)

Servicios suplementarios de completión de llamadas

Llamada en espera.(LLE)
 Retención de llamada.(RETEN)
 Completión de llamadas a abonado ocupado.(CLAO)

Servicios suplementarios pluripartitos.

Comunicación de conferencia.(CONF)
 Servicio tripartito.(3PTT)

Servicios suplementarios para comunidades de intereses

Grupo cerrado de usuarios.(GCU)
 Plan de numeración privado.(PNP)

Servicios suplementarios para tarificación

Llamadas con tarjetas de crédito.(CRED)
 Aviso del importe de la comunicación.(AIC)
 Cobro revertido.(REV)

Servicios suplementario de transferencia de información adicional.

Señalización de usuario a usuario.(SUU)

Como se puede apreciar de la lista anterior, el número de servicios disponibles, es muy grande y en un futuro muy cercano ésta lista irá creciendo para ofrecer otro tipo de servicios.

Esta gran ventaja de las centrales SPC, está basada precisamente en que todas las funciones de la central están almacenadas en la memoria del computador que controla la central y que el procesador administra.

La creación de esos programas de control de la central SPC, son de tipos muy variados y complicados, lo cual, hace necesario que el diseño de estos programas se haga por medio un método de diseño de software como veremos en el capítulo siguiente.

CAPITULO 2.- DISEÑO DE LA PROGRAMACION PARA UNA CENTRAL TELEFONICA DIGITAL

2.1 INTRODUCCION AL METODO DE TRABAJO PARA EL DISEÑO DE SOFTWARE.

Debido a que un sistema telefónico digital SPC será manejado por un gran número de gente durante todas sus fases desde el diseño e ingeniería pasando por producción y prueba hasta la instalación y puesta en operación, es necesario tener un sistema de diseño bien estructurado que facilite ese trabajo.

El diseño del software es un proceso muy complejo, ya que no solo es hacer un programa (ya sea en lenguaje de alto nivel ó en lenguaje ensamblador), el proceso del diseño del software involucra desde el estudio de los requerimientos ó factores que dan inicio al diseño del software hasta la elaboración y prueba del programa, así como también de la preparación de los documentos necesarios para soportar las decisiones técnicas tomadas y las estimaciones de costo del proyecto.

Este diseño del sistema deberá seguir un método de trabajo bien definido que cubra la sistematización, el diseño en sí, y la prueba, así como proveer un gran respaldo para asegurar la efectividad y la calidad del sistema.

EL objetivo de un método de trabajo es estructurar todo el diseño en actividades óptimas las cuales son fáciles de supervisar, dando por resultado tareas bien definidas y con una gran calidad.

Un sistema bien estructurado y con una meta fija es el pre-requisito para una buena calidad, gran productividad y entrega a tiempo del proyecto.

Generalmente el diseño de un sistema se divide en varias etapas que deberán ser llevadas en una secuencia debidamente planeada, cada etapa deberá estar bien definida con documentación de entrada y salida y con un claro límite de responsabilidad entre etapa y etapa lo cual facilita el flujo de información,

Una metodología a seguir sería la siguiente:

METODOLOGIA

PASOS FASES	DEFINICION DEL PROBLEMA	MEDICION DEL SISTEMA	ANALISIS DE DATOS	MODELADO DE SISTEMAS	SINTESIS DE SISTEMAS	TOMA DE DECISION
PLANEACION DE PROGRAMA	COMIENZO					
PLANEACION DEL PROYECTO						
DESARROLLO DEL SISTEMA						
PRODUCCION O CONSTRUCCION						
DISTRIBUCION O PUESTA EN SERVICIO						
OPERACION O CONSUMO						
RETIRO						FIN

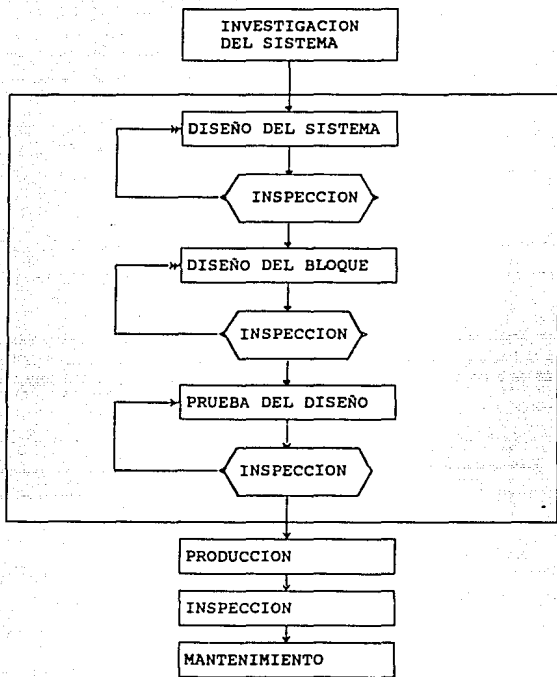
El resultado de una actividad después de haber sido aprobado, constituye la entrada de la siguiente etapa, cada etapa esta constituida por pequeñas actividades las cuales al ser finalizadas son inspeccionadas -lo cual redundante en la reducción significativa del número de fallas-, y aprobadas para completar así una etapa.

METODO A SEGUIR PARA EL DISEÑO DEL SOFTWARE

A continuación veremos como se estructuraremos el trabajo en el caso particular del diseño del software.

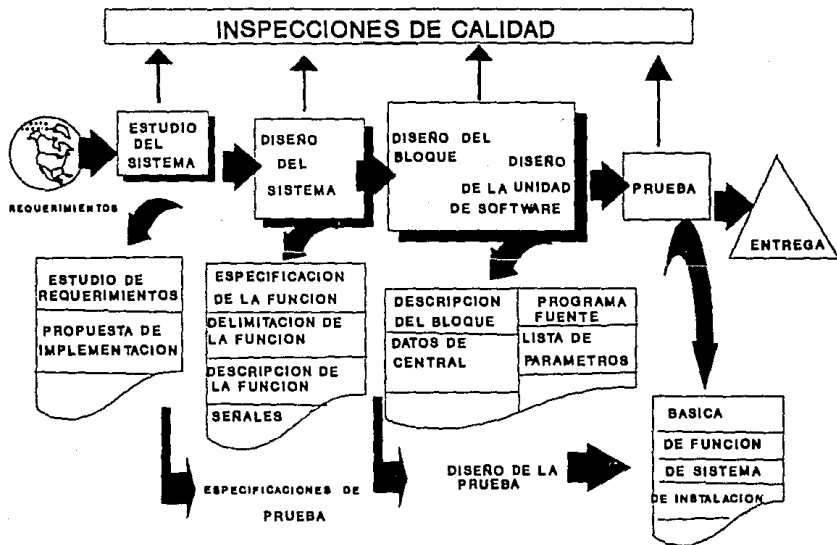
Cada una de las etapas del diseño se subdivide a su vez en otras, con el objeto de hacer más fácil y minucioso este diseño lo que permite tener un mejor control de el ya que al tener cada etapa bien definida se tiene el proyecto automáticamente bien definido.

Este será el método a seguir para el diseño de software y es conocido como método RUDOLPH el cual es uno de los más importantes en el mundo del diseño del software y se ha escogido en particular este método debido a la facilidad que se obtiene para controlar todo un proyecto de diseño de software, posteriormente se da una breve explicación de cada una de las etapas que componen este método, así como también de sus sub-etapas, y, en los capítulos III y IV se presenta un ejemplo real de la aplicación de este método, avalado por el programa fuente obtenido como resultado del proceso de este diseño y que se presenta como apéndice B.



Flujo de trabajo en el diseño del software

Metodo de diseño de Software



2.2 INVESTIGACION DEL SISTEMA

El trabajo de diseño es precedido por lo que se llama INVESTIGACION DEL SISTEMA, el que se divide en dos fases ESPECIFICACION DE LOS REQUERIMIENTOS y la PROPUESTA DE IMPLEMENTACION.

ESPECIFICACION DE LOS REQUERIMIENTOS

La especificación de los requerimientos es la fase del desarrollo del sistema, en el cual se establecen los principales requerimientos del sistema por parte del cliente, y esto es tomando en cuenta las siguientes referencias :

- Aplicaciones en la red - el cual establece para que tipo de central telefónica es requerido el sistema por ejemplo TANDEM, STP, SCCP, ETC.
- Especificaciones del cliente - por ejemplo las especificaciones de TELMEX.
- Otros Proyectos - por ejemplo pueden ser mejoras a un proyecto anterior ó adaptaciones de otros mercados.

PROPUESTA DE IMPLEMENTACION

Es iniciada a partir de nuevas asignaciones del departamento de planeación ó de las especificaciones de los requerimientos.

La función principal de ésta fase es hacer una analisis tanto del sistema como de las partes involucradas y/ó afectadas , así como mostrar la cantidad de trabajo, la división de este trabajo en varias etapas, también los volúmenes y costos de producción, además en está fase se pueden organizar las siguientes actividades :

- El ofrecimiento de nuevos servicios ó mejoras al cliente.
- Toma de decisión en la introducción de nuevos servicios.
- Verificar que los requerimientos sean suficientes y hayan sido correctamente interpretados.
- Revisar que la implementación propuesta corresponda con las reglas establecidas de antemano para el sistema.
- Un estimado de los recursos necesarios para el desarrollo del proyecto
- Planeación del proyecto

2.3 DISEÑO DEL SISTEMA

Aquí la función del sistema es estudiada y procesada a un nivel más bajo, en el cual puede ser dividido en uno ó varios bloques funcionales, la entrada a ésta fase son el ESTUDIO DEL SISTEMA, y la PROPUESTA DE IMPLEMENTACION debidamente inspeccionados y aprobados.

Esta etapa se subdivide en las siguientes fases :

- LIMITES DE LA FUNCION
- ESPECIFICACION DE LA FUNCION
- DESCRIPCION DE LA FUNCION
- INSTRUCCIONES DE OPERACION.

LIMITES DE LA FUNCION

Esta etapa nos sirve para describir brevemente como un requerimiento del cliente ó una mejora al sistema influye a otras partes del sistema total telefónico y tambien incluye una estimación de la cantidad de trabajo necesaria para llevar a cabo esa tarea. Cada requerimiento del cliente ó cada mejora del sistema deberá de ser tomado en cuenta durante ésta etapa, ya que aquí es donde se describen todos los cambios del sistema debido a las mejoras ó requerimientos de éste (cambios en funciones, en productos, señales, documentos, recursos, principios en el sistema, etc.), desde una muy bien definida base de diseño, éste documento es principalmente útil para las siguientes tareas:

-Diseño de bloques funcionales- para de ésta manera saber cual es la influencia de los cambios propuestos en el sistema.

-Actualizar la información de todas las señales que se usarán en el sistema y notificar a los usuarios de estas señales, de su modificación.

-Administración de recursos-para la etapa de diseño del bloque es importante conocer la cantidad de trabajo y así tener asegurada la gente necesaria para elaborarla.

Esta etapa es entendida como un breve sumario de la influencia del sistema comparada con la base del diseño, nuevas y modificadas funciones, bloques, relaciones señal-bloque, y documentos deberán ser especificados exactamente, una solución técnica breve deberá ser

descrita, solo para tener una idea de como se llevará a cabo la implementación.

ESPECIFICACIONES DE LA FUNCION

El objetivo principal de ésta fase es el de proveer una especificación de cada función dentro del sistema pero desde un punto de vista independiente de la propuesta de implementación, es decir, debe representar puramente la función en si, su manejo y sus parámetros asociados. Este documento es de gran importancia durante las siguientes actividades

-Presentación del sistema- Aquí la especificación de la función es usada para resaltar las características del sistema, y poder así, presentar al grupo técnico de evaluación del cliente una mejor idea de lo que se le esta ofreciendo.

-Análisis de oferta- La especificación de la función debe de ser escrita de tal manera que el cliente pueda fácilmente decidir si sus requerimientos son satisfactoriamente cubiertos.

-Trabajo de diseño del sistema- Este documento también sirve como un soporte para el trabajo del diseño del sistema, y debe de ser escrito en tal manera que ayude a la verificación de los requerimientos así como soportar la preparación para las pruebas del sistema.

DESCRIPCION DE LA FUNCION

Su principal objetivo es describir la función con énfasis en sus interacciones, hombre-máquina interfaces, estructuras internas, y la ubicación de producto dentro de la central.

INSTRUCCIONES DE OPERACION

Durante ésta etapa se describen una serie de fases de trabajo paso a paso, por lo tanto la instrucción de operación deberá de ser del tipo task-oriented, es decir orientada hacia el tipo de trabajo para la cual ésta escrita. Cada actividad independiente deberá tener una instrucción de operación de tal manera que le facilite al operador de la central el llevar a cabo su labor cuando requiere información de como llevar a cabo una tarea. Este documento es escrito para instalación, extensión y tareas administrativas en la central telefónica, y deberá ser almacenada en los manuales de mantenimiento.

2.4 DISEÑO DEL BLOQUE

Una vez que la función del sistema es estudiada y procesada a un nivel más bajo, se procederá a elaborar cada uno de los bloques funcionales.

Esta etapa se subdivide en las siguientes fases :

- INFORMACION SOBRE LA REVISION DEL PRODUCTO
- DESCRIPCION DEL BLOQUE
- DATOS DE CENTRAL

INFORMACION SOBRE LA REVISION DEL PRODUCTO

Esta es la primer fase, dentro de ésta etapa y brinda la información acerca de lo nuevo ó modificado en el bloque funcional a las personas responsables del diseño del sistema con la finalidad de tenerlo completamente actualizado.

También nos indica cuales son las modificaciones, es decir, el por que se inicia una nueva revisión del producto, con una base bien fundamentada, así como una solución técnica para cada problema ó modificación.

DESCRIPCION DEL BLOQUE

Durante esta fase, se describen todas las funciones del bloque funcional y muestra como es que esas funciones están distribuidas (hardware y software), también describe las interacciones con otros bloques funcionales, también usada para mostrar el resultado de la fase de diseño del bloque y sirve como base para comenzar la fase de del diseño de la unidad funcional de software.

Esta también es usada como una descripción del producto, para fines administrativos, para prueba del producto y como documento descriptivo para el cliente.

DATOS DE CENTRAL

Los datos de central también son estudiados durante la fase del diseño del bloque funcional, ya que es necesario que la central telefónica este cargada con estos datos, para poder así asegurar el funcionamiento de la misma.

2.5 UNIDAD DE SOFTWARE

La unidad de software de un bloque funcional es el programa que será almacenado en la memoria del procesador y que controlará las actividades de la central, es aquí precisamente de donde el sistema toma su nombre SPC sistema controlado por programa almacenado. La relación que el usuario tiene con el software de la central telefónica se muestra en la figura de la página siguiente.

La unidad de software consta básicamente de :

- PROGRAMA FUENTE
- LISTA DE PARAMETROS
- DATOS DE CENTRAL

PROGRAMA FUENTE

EL programa fuente es en si la implementación de la función por medio de un programa, que es escrito ya sea en lenguaje de alto nivel , ó bien en lenguaje ensamblador cuando se desea tener un control más estricto de las direcciones en al memoria. Este programa deberá de estar bien estructurado de tal manera que sea fácil de dar mantenimiento.

LISTA DE PARAMETROS

Un producto para poder ser competitivo en el mercado, deberá de tener una gran flexibilidad, es aquí donde radica la importancia de ésta fase, ya que es aquí precisamente donde se establecen los parámetros que regirá el programa fuente y que son netamente dependientes del mercado, datos que al ser adecuadamente modificados hacen que el mismo programa fuente sirva tanto para un mercado como para otro.

2.6 INSPECCIONES

Una de las formas de asegurar la calidad del sistema y además una de las formas más importantes de evitar pérdida de tiempo debido a problemas fuertes, es la etapa de inspección.

Más que una etapa ó fase , la inspección es un procedimiento que deberá de realizarse al final de cada una de las fases ó etapas del proyecto y que además no se podrá avanzar a la siguiente etapa ó fase del proyecto, si durante la inspección final de la fase actual se

detectan errores ó fallas y no se cumple con los requisitos minimos para pasar su aprobación.

2.7 PRUEBA

Una vez que el diseño esta hecho, deberá ser probado a fin de constatar que los requerimientos han sido totalmente cubiertos, el proceso de prueba esta dividido en varias etapas con diferentes objetivos y diferentes enfoques, es decir cada fase de prueba es diferente entre si, con la finalidad de probar el sistema totalmente desde puntos de vista diferentes. También es necesario mencionar que la prueba del sistema ayuda a reducir el factor de riesgo ó número de fallas del sistema, pero en ningún momento, la prueba hace ó mejora la calidad del sistema.

La idea principal de una prueba de un sistema de software es tener el mayor número de casos de prueba -por línea de código del programa- en la fase básica de prueba y el menor número de casos de prueba por línea de código en las fase de prueba más alta.

Para cada una de estas pruebas se elaboran documentos llamados *ESPECIFICACIONES DE PRUEBA* e *INSTRUCCIONES DE PRUEBA*, en el primero se especifican todas las pruebas a que el producto será sometido y en el segundo documento se indica la forma en que estas pruebas se llevarán a cabo, y generalmente éste tipo de pruebas son casi en su totalidad automatizadas.

Las fases de la prueba las podemos dividir en las siguientes:

PRUEBA BASICA

PRUEBA FUNCIONAL

PRUEBA DE FABRICA

PRUEBA DE SISTEMA FUENTE

PRUEBA DE INSTALACION

DEMOSTRACION DEL SISTEMA DE APLICACION

PRUEBA BASICA

Aquí cada unidad es probada por separado, es una verificación de la fase de diseño a nivel de código para

ver que el programa cumple con los requerimientos y reglas especificados en el documento *DESCRIPCION DEL BLOQUE*. Esta prueba se hace a nivel de PC computadora personal.

PRUEBA FUNCIONAL

Esta fase es donde se prueban las funciones primero de una manera aislada y luego es conjunto, de tal manera que se cumpla con los requisitos funcionales e interfaces entre hardware y software, esta prueba es en si una de las más importantes fases, ya que es aquí donde la función es totalmente probada, para saber que los requerimientos por parte del cliente han sido satisfactoriamente cumplidos. Esta prueba se realiza en centrales telefónicas de prueba y con ayuda de simuladores

PRUEBA DE FABRICA

Esta prueba es realizada a nivel de hardware y es con la finalidad de verificar que todo el equipo esta trabajando en perfectas condiciones incluyendo los nuevos requerimientos de software de operación y mantenimiento.

PRUEBA DE SISTEMA FUENTE

Es es realizada ya en el "campo", es decir, en una central telefónica que pertenece al cliente y la cual es manajada por gente que ha sido totalmente ajena al proceso de diseño, éste tipo de prueba incluye un variado número de casos de prueba pero no es tan profunda en el sentido de los requerimientos, ya que más bien esta prueba esta enfocada para comprobar la capacidad y la exactitud del sistema.

PRUEBA DE INSTALACION

Aquí los datos de la central así como los de tráfico, hardware, software y cableado son totalmente probados, ya que para la aceptación del sistema, esta prueba debe de ser aprobada antes de ser entregado al cliente el producto. En éste nivel los casos de prueba son diseñados desde el punto de vista operativo y de instalación.

DEMOSTRACION DEL SISTEMA DE APLICACION

Esta es la primer prueba que se realizará en una central telefónica real y es con la finalidad de mostrar el sistema ya en operación al cliente. El documento instrucciones de operación es el principal documento usado en esta prueba, y una variedad de casos de prueba son seleccionados sólo a manera de demostración.

2.8 LIBERACION DEL PRODUCTO

Finalmente cuando el producto ha sido totalmente terminado y probado satisfactoriamente, puede ser liberado es decir entregado al cliente, pero para poder llevar a cabo éste proceso es necesario elaborar antes los reportes de calidad del software el requerimiento de liberación en los cuales se especifica la calidad del producto, con sus características al momento de terminarse, así como una explicación en el caso de que no se haya seguido el flujo de trabajo y se afectará la calidad del producto.

CAPITULO III DISEÑO DEL SISTEMA

3.1 LIMITES DE LA FUNCION "LLAMADA EN ESPERA"

DESCRIPCION

LLAMADA EN ESPERA (CAW) es un servicio suplementario que permite a un abonado ser notificado de la presencia de una llamada entrante (según los procedimientos de llamadas básicas) mediante una indicación de que ningún canal de información del interfaz esta disponible. El usuario podrá entonces elegir entre aceptar, rechazar o ignorar la llamada en espera (según los procedimientos de llamadas básicas).

CLASE

Clase: Opcional, dependiente del mercado.

Para uso en centrales telefónicas Públicas.

SOLUCION TECNICA

INFORMACION GENERAL

La facilidad de usuario LLAMADA EN ESPERA, será implementada dentro del "Sistema de administración de tráfico" (TMS) .

La facilidad de usuario LLAMADA EN ESPERA indica, por medio de un tono especial de audio llamado "tono de llamada en espera", a un suscriptor que esta haciendo uso del servicio telefónico (suscriptor ocupado), que otra llamada esta esperando a ser contestada en la misma linea.

El suscriptor ocupado puede entonces, ignorar el tono y seguir con la llamada original, ó contestar la llamada que esta en espera.

Para contestar la llamada que esta en espera el susbscriptor ocupado, puede usar dos métodos :

- 1.- Conmutando es decir por medio de bajar y subir la horquilla de su aparato telefónico con un movimiento rápido (cuelga/descuelga).

De ésta manera el subscriptor que estaba en espera, ahora esta en linea y puede comunicarse con el subscriptor que estaba ocupado, y el subscriptor que inicialmente estaba en comunicación con el usuario de la facilidad, ahora es el subscriptor en espera. El subscriptor que tiene esta facilidad de usuario puede conmutar las veces que deseé y así hablar con uno ú otro de los subscriptores.

- 2.- Terminado la llamada, es decir, cuelga su aparato telefónico, con lo que éste subscriptor queda reservado y es retollamado por la llamada que estaba en espera, es decir el teléfono del subscriptor sonará indicando que tiene una llamada, a al contestar, es conectado automáticamente al subscriptor que está en espera.

Si el subscriptor tiene también contratada la facilidad de usuario Retención de la llamada, entonces al momento de hacer la conmutación para contestar la llamada en espera, tendrá que marcar un código predeterminado para regresar a la llamada que tiene retenida.

La facilidad de usuario LLAMADA EN ESPERA es asignada a un subscriptor por medio del comando SUSCC y puede ser activada ó desactivada por medio del comando SUFIC. El comando SUFIC también puede ser usado para dar al subscriptor acceso a un procedimiento del subscriptor, es decir un procedimiento por medio del cual él puede activar ó desactivar ésta facilidad de usuario.

Proveer:

SUSCC:SNB=snb,SCL=CAW-1;

Remover:

SUSCC:SNB=snb,SCL=CAW-0;

Activar:

SUFIC:SNB=snb,FEATURE=CAW,ACT,SCI=(ACT,NONE, ó STD);

Deactivar:

SUFIC:SNB=snb,FEATURE=CAW,DEACT,SCI=(ACT,NONE, ó STD);

LLAMADA EN ESPERA ENTRANTE es una versión de la facilidad LLAMADA EN ESPERA y trabaja de la misma forma que se menciona arriba, la única diferencia es que ésta facilidad acepta solo llamadas en espera pero dentro del mismo grupo de distribución de una de una central telefónica, para llamadas provenientes de otra central telefónica, es decir, de otro grupo de distribución, la llamada recibe el tratamiento de "OCUPADO".

Esta facilidad de usuario también usa el comando BGSCC y SUFIC para fines administrativos.

Proveer:

BGSCC:BG=bg,BSNB=bsnb,SCL=CWI-1;
BGSCC:BG=bg,BGS=bgs,SCL=CWI-1;

Remover:

BGSCC:BG=bg,BSNB=bsnb,SCL=CWI-0;
BGSCC:BG=bg,BGS=bgs,SCL=CWI-0;

Activar:

SUFIC:SNB=snb,FEATURE=CWI,ACT,SCI=(ACT,NONE ó STD);

Deactivar:

SUFIC:SNB=snb,FEATURE=CWI,DEACT,SCI=(ACT,NONE ó STD);

FUNCIONES INFLUENCIADAS

Especificación de funcionamiento de LLAMADA EN ESPERA
Especificaciones de CATEGORIAS DE SUBSCRIPTOR
Especificaciones de PRUEBA LLAMADA EN ESPERA

SUBSISTEMAS INFLUENCIADOS

SUS

SUSCAW - Un nuevo bloque controlará el tráfico telefónico cuando LLAMADA EN ESPERA este en uso. SUSCAW también analizará los datos recibidos cuando el subscritor intente modificar el servicio de llamada en espera a través de un procedimiento de subscritor.

TCS

Una nueva categoría (CWI) es requerida para LLAMADA EN ESPERA ENTRANTE, que es una porción de la facilidad de usuario LLAMADA EN ESPERA. Para permitir que la interacción entre las facilidades de subscritor LLAMADA EN ESPERA y RETENCION DE LLAMADA, el bloque funcional SFSHD deberá de trabajar de la siguiente manera :

Si un subscritor con ambas facilidades de subscritor se encuentra ocupado y recibe una llamada entonces SFSHD se encargará exclusivamente del establecimiento de la llamada después, SUSCAW hará el análisis de ésta y mandará los tonos, después de esto el control de la llamada es cambiado nuevamente a SFSHD y éste se

encargará de el tráfico hasta que la llamada es terminada.

OIS

Este subsistema es impactado debido al cambio de los comandos SUSCC, SUFIC, BGSCC, usados para la administración de la facilidad de usuario, el impacto exácto deberá de ser determinado por el responsable del subsistema OIS.

MMS

Este subsistema es impactado debido al cambio de los comandos SUSCC, SUFIC, BGSCC, usados para la administración de la facilidad de usuario, el impacto exácto deberá de ser determinado por el responsable del subsistema MMS.

INFLUENCIA EN LA INTERFACE SISTEMA/SUBSISTEMA

Ningún tipo de influencia es esperada.

INFLUENCIA EN LOS RECURSOS DEL SISTEMA.

La carga del procesador no deberá de rebasar los límites establecidos para un bloque de tráfico.

PRINCIPIOS DEL SISTEMA QUE SON INFLUENCIADOS

Ningún tipo de influencia es esperada.

COSTO DE LA IMPLEMENTACION**ELABORACION DEL BLOQUE FUNCIONAL Y DISEÑO DE LA UNIDAD FUNCIONAL DE SOFTWARE**

SUBSISTEMA	BLOQUE	INSTRUCCIONES DE PLEX	MANO DE OBRA (EN HORAS- HOMBRE)
-----	-----	-----	-----
SUS	SUSCAW	800-1000	500-600
TCS	SFSDH	1000-1200	600-700
OIS	El impacto no ha sido determinado aún.		
MMS	El impacto no ha sido determinado aún.		

PRUEBA FUNCIONAL

FUNCION	MANO DE OBRA (EN HORAS HOMBRE)
-----	-----
LLAMADA EN ESPERA	300-400

3.2 ESPECIFICACIONES DE FUNCIONAMIENTO DE "LLAMADA EN ESPERA"

PROPOSITO

La función de la facilidad de usuario "LLAMADA EN ESPERA" es la de informar a un subscriptor que se encuentra ocupado haciendo uso de su línea telefónica, que otra llamada está esperando a ser contestada en esa misma línea, el subscriptor puede entonces:

- a) Ignorar el tono y continuar con la llamada original.
- b) Retener la llamada original y conmutar para contestar la nueva llamada (llamada en espera).
- c) Colgar, y de esta manera termina con la primer llamada, queda libre y es llamado nuevamente, pero ahora por el subscriptor el cual estaba en espera

La facilidad de usuario llamada "LLAMADA EN ESPERA ENTRANTE" trabaja en la misma forma que la facilidad "LLAMADA EN ESPERA", la única diferencia es que ésta facilidad solo trabaja para usuarios dentro del mismo grupo de distribución de la central telefónica, es decir si la llamada viene de otro grupo ó central telefónica diferente la llamada recibe un tratamiento de ocupado.

SUBFUNCIONES

- a) Provisión .-El acceso es provisto por medio de comandos en la central telefónica.
- b) Modificación .- La modificación es provista por medio de comandos en la central telefónica.
- c) Cancelación .- Acceso es cancelado por medio de comandos en la central telefónica.
- d) Activación. La facilidad es activada por medio de marcar un código desde el aparato telefónico,

ó bien por medio de un comando desde la central telefónica.

- e) Desactivación. La facilidad es desactivada por medio de marcar un código desde el aparato telefónico, ó bien por medio de un comando desde la central telefónica.
- f) Control del tráfico de la central telefónica cuando la facilidad esta en uso.

TABLA DE ACCIONES DE LOS SUBSCRIPTORES

La tabla siguiente muestra las posibles acciones de los subscriptores bajo varios estados, con las indicaciones de que hacer ó a donde dirigirse en esas circunstancias, para el mejor entendimiento de ésta tabla favor de referirse a las definiciones de los diferentes símbolos.

DEFINICIONES

CON	Conexión normal entre dos subscriptores sin tener la facilidad de usuario en uso.
NA	No Aplica, combinación invalida al hacer uso de ésta facilidad de usuario.
RB	Retrollamada al subscriptor.
RDT	Tono de marcar.
TO	Tiempo de supervisión expirado.
TS	Tiempo de supervisión.

Para mayor claridad en éste documento, los tres subscriptores son denominados A, B, C.

A y B representan una llamada normal entre dos subscriptores. C representa la llamada en espera.

A es el subscriptor que tiene la facilidad de usuario y puede ser el subscriptor que llamo originalmente a B ó el subscriptor que es llamado por el subscriptor B.

A	-----+	
		llamada Normal
B	-----+	
C	-----	llamada en espera

Si el subscriptor B es el que originó la llamada, es decir B llamo a A, entonces no se aplica el tiempo de supervisión cuando éste cuelga, si no que es desconectado inmediatamente.

La siguiente tabla no tiene ninguna relación con el tiempo de duración de los tiempos de supervisión dado que éste tiempo no es medido por la facilidad de usuario.

Los procedimientos aquí descritos son los únicos que son permitidos durante el uso normal de la facilidad de usuario "LLAMADA EN ESPERA"

ESTADO DE LA LLAMADA

	1. A-B hablando CAW tono hacia A	2. A-C hablando B Retenido	3. A-B hablando C Retenido	4. A-B hablando CAW tono a A. B en TS	5. A-B hablando C Retenido B en TS	6. A-B hablando B en TS	
A C C I O N E S	A Conmuta	Pasa a 2 *NOTA1*	Pasa a 3 *NOTA1*	Pasa a 2 *NOTA1*	desc B A-C CON	desc B A-C CON	No cambio
	A Cuelga	Pasa a 7 *NOTA2*	Pasa a 8	Pasa a 7	Pasa a 7 *NOTA2*	Pasa a 7	desc A, B
D E L	B Cuelga	Pasa a 4	desc B A-C CON	Pasa a 5	NA	NA	NA
	C Cuelga	desc C A-B CON para CAW	desc C A-B CON	desc C A-B CON	desc C para CAW Pasa a 6	desc C Pasa a 6	NA
S U B S C R I P T O R	B Contesta	NA	NA	NA	Pasa a 1	Pasa a 3	A-B CON
	TS expira para B	NA	NA	NA	desc B A-C CON	desc B A-C CON	desc A, B
	C Contesta	NA	NA	NA	NA	NA	NA
	TS expira para C	NA	NA	NA	NA	NA	NA

ACCION O CONDICION RESULTANTE

NOTA1 - Si el subscritpor A tiene también la facilidad de usuario "Retención de llamada", cuando el conmuta para contestar la llamada en espera, recibirá un tono de marcar y en ese instante deberá marcar el código de Retención de llamada antes de moverse al siguiente estado.

NOTA2 - Si el subscriptor A debe colgar ó cuelga antes de contestar la llamada en espera, su teléfono sonará nuevamente hasta contestar la llamada que está en espera.

ESTADO DE LA LLAMADA

		7. A es llamado por C, B desc.	8. A es llamado por B C desc.			
A C C I O N	A Consulta	NA	NA			
	A Cuelga	NA	NA			
D E L	B Cuelga	NA	desc A, B, para tono			
	C Cuelga	desc A, C. para tono	NA			
S U B S C R I P T O R	A contesta	A-C CON	A-B CON			
	B contesta	NA	NA			
T O R	TS expira para B	NA	NA			
	C contesta	NA	NA			
	TS expira para C	NA	NA			
	Retrolla mada	desc A, C	desc A, B			

ACCION O CONDICION RESULTANTE

CONDICIONES DE OPERACION**EQUIPO DEL SUBSCRIPTOR**

El subscritor podrá usar un DTMF ó un teléfono con disco para éste procedimiento

EQUIPO DE LA CENTRAL TELEFONICA.

Dispositivos con circuitos multiconferencia ó un puerto multi-conector son necesarios para el funcionamiento de ésta facilidad de usuario.

PROCEDIMIENTOS OPERACIONALES**COMANDOS DE OPERACION DE LA CENTRAL**

La facilidad es provista y dada de alta por medio de comandos desde la central telefónica.

La activación y desactivación de la facilidad de usuario es dada de alta por medio de comandos desde la central telefónica.

Un subscritor llamado TERMINAL 1 puede ser asignado por comando desde la central telefónica.

La información para el control del subscritor normal (SCI) y para el control de subscritor terminal 1 (SCI1) pueden ser cambiadas por medio de comandos desde la central telefónica.

Los valores para éste parámetro de ACT, STD y NONE determinan el tipo de procedimientos que el subscritor podrá ejecutar desde su aparato telefónico.

NONE significa que el subscritor no podrá realizar ningún tipo de procedimiento desde su aparato, solo podrá utilizar la facilidad de usuario.

STD significa que el subscritor ésta permitido ó no a usar los procedimientos (es decir esto depende del estandar usado en el mercado, valor dependiente de mercado).

ACT significa que el subscritor puede realizar procedimientos de subscritor desde su teléfono, es decir, activar, desactivar ó seleccionar su terminal 1.

Estos parámetros podrán ser cambiados desde la central telefónica por medio del comando SUFIC.

PROCEDIMIENTOS DE SUBSCRIPTOR

ACTIVACION/DESACTIVACION DE LA FACILIDAD DE USUARIO
LLAMADA EN ESPERA

El subscritor podrá (de acuerdo al valor del parámetro SCI) activar ó desactivar la facilidad de usuario por medio de un "Procedimiento de subscritor".

Un subscritor Terminal 1 es un subscritor que tiene la capacidad de Activar ó desactivar la facilidad de usuario Llamada en Espera de otros subscritores. Un subscritor con la facilidad de llamada en espera podrá tener solo un subscritor Terminal 1.

Cuando la facilidad esté desactivada ninguna llamada en espera será permitida y el subscritor con llamada en espera recibirá tono de ocupado, la facilidad permanecerá desactivada hasta que sea activada nuevamente por medio de comando en la central ó por medio de un procedimiento del subscritor.

Los siguientes procedimientos son usados por el subscritor para activar ó desactivar la facilidad de usuario LLAMADA EN ESPERA.

Procedimiento de Activación:
(SCP) SCA SCS CT

Procedimiento de Desactivación:
(SCP) SCD SCS CT

Los siguientes procedimientos son usados por el subscritor Terminal 1 para activar ó desactivar la facilidad de usuario LLAMADA EN ESPERA de otros subscritores.

Procedimiento de Activación:
(SCP) SCA SCS RDT SNB CT

Procedimiento de Desactivación:
(SCP) SCD SCS RDT SNB CT

Notaciones usadas:

CT = Tono de Confirmación.

RDT = Tono de retrollamada.

SCA = Código de servicio para activación de la facilidad.

SCD = Código de servicio para desactivación de la facilidad.

SCP = Código de servicio prefijo (* ó 11).

SCS = Código de servicio sufixo (# ó expiración de tiempo).

SNB = Numero del subscritor que contiene la facilidad a ser cambiada.

CONTESTACION DE LA LLAMADA EN ESPERA**SUBSCRIPTOR SOLO CON "LLAMADA EN ESPERA"**

Un subscritor solo con la facilidad de "LLAMADA EN ESPERA" puede contestar la llamada de las siguientes formas:

- 1 Conmutando, es decir por medio de colgar y descolgar rápidamente su teléfono podrá conectarse a la llamada en espera, poniendo a subscritor de la llamada original en Retención. El subscritor podrá conmutar entre los subscritores B y C por las veces que el lo deseé.
- 2 Colgando, con lo cual termina con la llamada original, es decir el subscritor B es desconectado y entonces al quedar el subscritor A libre, es llamado automáticamente por el subscritor C por lo tanto A y C serán conectados hasta que A conteste y se haga cargo de la llamada.

SUBSCRIPTOR CON LAS FACILIDADES DE "LLAMADA EN ESPERA" Y "LLAMADA EN RETENCION"

Si el subscritor tiene las dos facilidades contratadas, "LLAMADA EN ESPERA" y "LLAMADA EN RETENCION", entonces el siguiente procedimiento es seguido para contestar la "LLAMADA EN ESPERA" o conmutar entre la llamada original y la que esta en espera.

- 1 Colgando, con lo cual termina con la llamada original, es decir el subscritor B es desconectado y entonces al quedar el subscritor A libre, es llamado automáticamente por el subscritor C por lo tanto A y C serán conectados hasta que A conteste y se haga cargo de la llamada.
- 2 FLA RDT SCH

Notaciones:

FLA = Conmutación realizada por el subscritor

con la facilidad.
 RDT = Tono de retrollamada.
 SCH = Código de servicio para la facilidad de Retención de llamada.
 No existe límite en el número de veces que el suscriptor conmute de uno a otro usando este procedimiento.

CATEGORIAS DEL SUBSCRIPTOR

La función de "LLAMADA EN ESPERA" requiere que el suscriptor tenga asignada la categoría CAW-1 para tener acceso a la facilidad "LLAMADA EN ESPERA".

Para la facilidad de "LLAMADA EN ESPERA ENTRANTE" se requiere de la categoría CWI-1.

TASACION

En la CAW/CWI configuración existen dos llamadas que deberán ser tasadas, la llamada original (A-B) y la nueva llamada entrante ó contestación de llamada en espera (A-C) la tasación se realiza normalmente en las dos llamadas sin importar quien las haya originado y sea el controlador de ellas.

Los procesos de tasación para la llamada en espera comienzan cuando el suscriptor que tiene la facilidad de "LLAMADA EN ESPERA" contesta al suscriptor que tiene en espera, ya sea por medio de colgar ó por medio de conmutar.

Los procesos de tasación normalmente terminan cuando el suscriptor que originó la llamada cuelga, si el suscriptor que estaba siendo llamado cuelga, la tasación se detiene en ese punto, y en el caso de que el suscriptor éste retenido la tasación continuará hasta que el controlador de la llamada conmute con lo cual la tasación continúa hasta que uno de los dos cuelgue, ó que el tiempo de supervisión para el suscriptor que está retenido en cuyo caso la tasación se detiene para contestar.

Cargos adicionales se podrán hacer a los suscriptores por hacer uso de los procedimientos de suscriptor (activación, desactivación, selección de un terminal 1) así como también podrán hacer cargos adicionales si el suscriptor con la facilidad de "LLAMADA EN RETENCION" usa un código de Retención para contestar la llamada en espera.

DISPONIBILIDAD

La facilidad de "LLAMADA EN ESPERA" es en principio disponible para todas las líneas de suscriptor sencillas (single party line), esto incluye líneas individuales dentro de una PBX, MLHG ó grupos de distribución especiales.

La facilidad de "LLAMADA EN ESPERA ENTRANTE" esta disponible para todas las líneas individuales dentro de un grupo especial de distribución, pero ésta facilidad no podrá ser asignada a una línea en una PBX ó MHLG.

3.3 DESCRIPCION DE LA FUNCION LLAMADA EN ESPERA**INTRODUCCION**

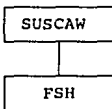
La función de la facilidad de usuario LLAMADA EN ESPERA es la de informar, por medio de un tono especial a un suscriptor que se encuentra ocupado, que otra llamada esta esperando a ser contestada.

UBICACION

La función de LLAMADA EN ESPERA se encuentra alojada en el bloque funcional SUSCAW dentro del subsistema del suscriptor SUS, el unico bloque con una interface hacia SUSCAW es FSH del Subsistema de Control de Tráfico (TCS).

SUSCAW - controla el manejo del tráfico cuando LLAMADA EN ESPERA está en uso. SUSCAW también analiza los datos recibidos cuando algún suscriptor intenta modificar su servicio de LLAMADA EN ESPERA a través de algún procedimiento de suscriptor.

FSH - Realiza las tareas ordenadas por SUSCAW (tareas como envío de información lectura de datos, manipula las llamadas etc.)

DIAGRAMA DE BLOQUES

SEÑALES

Esta es una lista de las señales que SUSCAW usa durante el manejo de tráfico y el procesamiento de las llamadas. El uso de números para cada señal es simbólico y es utilizado para cada caso de manejo de tráfico así como también permite simplificar el manejo de las tablas de estados que aparecen en este mismo documento.

La simbología de estos números es como sigue:

- a) Si un número es asignado a una señal, el número como tal indica la transmisión de dicha señal. por ejemplo '9' significa que el bloque funcional recibe la señal ProvideInstruc, y por ejemplo '7' significa que el bloque funcional envía la señal Finished.
- b) Si un número es asignado a un par de señales, este número es usado para representar la solicitud de datos así como la respuesta a ésta solicitud, por ejemplo '4' significa que Split es enviada y que SplitR es recibida como respuesta.

ADMINISTRACION DE LAS INTERCONEXIONES

1	Create/Creator
2	Free/FreeR
3	Join/JoinR
4	Split/SplitR

ADMINISTRACION DE LOS RECURSOS

5	LocResources/LocResourcesR
6	RelResources/RelResourcesR

ADMINISTRACION DEL DIALOGO

7	Finished
8	Handover/HandoverR (Chained)
9	ProvideInstruc
10	RelControl/RelControlR

ADMINISTRACION DE EVENTOS

11*	ClearTimer/ClearTimerR
12	Event
13*	GetTimer/GetTimerR
14	MonitorEvent

15* PostEvent/PostEventR
 16 ReadyForEvent
 17* SetTimer/SetTimerR

ADMINISTRACION DE LA INFORMACION

18 RetrieveInf/RetrieveInfr
 22 Commit/CommitR
 23 DeleteInf/DeleteInfr
 24 InsertInf/InsertInfr
 25 Prepare/PrepareR
 26 Rollback/RollbackR
 27 InicioTransact/InicioTransactR
 28 UpdateInf/UpdateInfr
 29 CreateTable/CreateTableR
 30 CreateField/CreateFieldR

TRANSFERENCIA DE INFORMACION A LA RED

19 ReceiveInf/ReceiveInfr
 20 SendInf/SendInfr
 21 SendReceiveInf/SendReceiveInfr

* NOTA: Estas señales no son usadas por SUSCAW, sin embargo se mencionan en éste documento para mantener el orden numerico.

CARGA INICIAL Y RECARGA DEL SISTEMA

Cuando el sistema es cargado por primera vez ó bien cuando es recargado, SUSCAW le indica en que fases de éste arranque intervendra, de la siguiente manera :

Fase 1 Fase absoluta 21, SUSCAW carga la referencia del bloque
 Fase 2 Fase absoluta 24, SUSCAW envia las señales FEATURETEXT con los datos de LLAMADA EN ESPERA , y también durante ésta fase crea las tablas donde cada subscriptor que tiene ésta facilidad es registrado.
 Fase 4 Fase absoluta 29, hace un rastreo para conocer el estado de cada llamada.
 Fase 7 Fase absoluta 44, SUSCAW envia la señal DIDINFO con datos de contadores para fines estadísticos.

TABLAS DE DATOS DE LLAMADA EN ESPERA

La facilidad de LLAMADA EN ESPERA interactua con varias bases de datos por medio las las señales de TCS mencionadas arriba.

La tabla de datos "Propiedades del subscriptor" contiene información acerca de cada subscriptor, ésta definida en el sistema y es actualizada cada vez que la facilidad es asignada ó removida a algún subscriptor.

La tabla de Disparo contiene información acerca de cuales subscriptores tienen la facilidad de usuario activada ó desactivada ésta tabla también esta definida por el sistema y es actualizada cada vez que la facilidad es activada ó desactivada.

La tabla LLAMADA EN ESPERA es creada por SUSCAW y contiene información para cada subscriptor que tiene la facilidad.

Esta tabla esta definda de la siguiente manera :

SC- Apuntador	SCI	SCI1	SC- Apuntador terminall	Terminal1 contador
------------------	-----	------	-------------------------------	-----------------------

- SC-apuntador - Apuntador del subscriptor, llave de entrada para la tabla.
- SCI - Información para el control del subscriptor dado por el comando SUFIC e inicialmente con el valor de STD
- SCI1 - Información para el control del subscriptor Terminal 1 dado por el comando SUFIC inicialmente con el valor de NONE
- SC1-apuntador - Apuntador del subscriptor Terminal 1, inicialmente puesto a NULL
- Terminal 1 - Contador de subscriptores terminal 1 inicialmente puesto a 0

PROVISION DE LA FACILIDAD DE USUARIO

Cuando CAW/CWI es asignada a algún subscriptor, por medio del comando SUSCC, SUSCAW es informado por medio de la señal ProvideInstruc. Cada vez que la facilidad es asignada, SUSCAW obtiene un SC-apuntador por medio de la señal ReceiveInf/R.

Si no existe ningún registro en las tablas de SUSCAW para éste subscriptor, se crea un nuevo registro por medio de las señales InsertInf/R

La tabla de disparo es actualizada.

Las tablas de LLAMADA EN ESPERA son actualizadas.

CANCELACION DE LA FACILIDAD DE USUARIO

Cuando la facilidad de usuario CW/CWI es removida de algún subscriptor por medio del comando SUSCC, SUSCAW es informado por medio de la señal ProvideInstruc.

SUSCAW obtiene el SC-apuntador por medio de ReceiveInf/R.

Los datos son borrados de la tabla de LLAMADA EN ESPERA por medio de la señal DeleteInf, a menos que el subscriptor este provisto con un terminal 1, ya que en éste caso primero es removido el terminal 1 y después el subscriptor con la facilidad de usuario.

La tabla de disparo es actualizada.

Las tablas de LLAMADA EN ESPERA son actualizadas, así como las bases de datos, finalmente el susbscriptor es liberado nuevamente.

METODOS DE ACTIVACION/DESACTIVACION

Cuando la facilidad de usuario esta desactivada ninguna LLAMADA en espera es permitida. Cuando la facilidad es activada entonces una y solo una llamada en espera es permitida al mismo tiempo.

Existen tres métodos de activar ó desactivar la facilidad:

- (a) Por medio del comando SUFIC.
- (b) Por medio de un procedimiento del subscriptor, lo que significa que el propio subscriptor desde su aparato telefónico por medio de marcar un código preestablecido activa ó desactiva la facilidad.
- (c) Por medio de un procedimiento del subscriptor TERMINAL 1, lo que significa que el propio subscriptor puede designar a otra persona (por ejemplo a su secretaria) y ésta otra persona

desde su aparato telefónico por medio de marcar un código preestablecido activa ó desactiva la facilidad, para el subscriptor.

CAMBIOS EN LA FACILIDAD DE USUARIO

Los parámetros de la facilidad de usuario, pueden ser modificados desde la central telefónica por medio de comandos (SUFIC) datos tales como quien es el Terminal 1 del subscriptor ó cuales son los valores que el subscriptor tendrá cuando se le asigne la facilidad, si el ésta ó no habilitado para activar ó desactivar la facilidad, todos estos cambios son llamados cambios administrativos y son realizados desde la central telefónica exclusivamente.

MANEJO DEL TRAFICO

Comienza cuando algún subscriptor intenta hacer una llamada a un subscriptor que tiene la facilidad LLAMADA EN ESPERA activada y el ésta ocupado en otra conversacion.

En éste momento por medio de señales el bloque SUSCAW comienza a interactuar con el bloque FSH revisando todo tipo de información, y actualizando las bases de datos, también pone una marca especial en esa llamada para de éste modo saber que se está haciendo uso de ésta facilidad en éste momento.

Por medio de ésta señalización SUSCAW ubica a los dos subscriptores A y B así com también checa sus categorías es decir que por ejemplo A tenga la facilidad provista y activada. También el subscriptor C es ubicado y checado para en el caso de que si el abonado A tiene solo la facilidad de LLAMADA EN ESPERA ENTRANTE asignada y el subscriptor C es un subscriptor de otra central publica, la LLAMADA EN ESPERA no es aceptada.

Una vez que estas pruebas han sido aceptadas, SUSCAW envía el tono de LLAMADA EN ESPERA al subscriptor A y la señal de llamando al subscriptor C, a partir de éste momento sea quien sea el que haya originado la llamada no importa, ya que apartir de éste momento el subscriptor A que es el que tiene la facilidad asignada será el controlador de la llamada.

TABLAS DE ESTADO DE LA LLAMADA, ACCIONES DEL SUBSCRIPTOR

Las tablas en las páginas siguientes muestran las acciones de cada uno de los subscriptores ante cada una de las posibles acciones que el pudiera realizar en cada momento de la llamada.

Definiciones

CAWT	Tono de llamada en espera
CI	Imagen de la llamada.
CON	Conexión normal entre dos abonados
NA	Combinación No Aceptada en ésta facilidad.
RB	Retrollamada al subscritor
TS	Desconexión del tiempo de supervisión.
TO	Tiempo de supervisión expirado.
CIA	Imagen de la llamada ha sido alterada.

		ESTADO DE LA LLAMADA			
		1.	2.	3.	4.
		A-B hablando C llamando CI - I	A-C hablando B retenido CI - II	A-B hablando C retenido CI - I	A-B hablando B en TS C llamando CI - I
A C C I O N E S	A conmuta	12,20a,20a 4,3,14,16 pasa a 2	12,4,3,14 16 Pasa a 3	12,4,3,14 16 Pasa a 2	12,20a, 20a,4,2 3,10,7 Desc B A-C CON
	A Cuelga	12,4,2,2, 20a,1b, 14,16 Desc B Pasa a 9	12,4,2, 20b,14,16 Desc C Pasa a 7	12,4,2, 20b,14,16 Desc B Pasa a 6	12,4,2,2 20a,1b, 14,16 Desc B Pasa a 9
D E L	B Cuelga	12,16 Pasa a 4	12,2,10,7 desc B A-C CON	12,16 Pasa a 5	NA
S U B S C R I P T O R	C Cuelga	12,20a, 2,10,7 Desc C Para CAWT	12,4,2,3, 10,7 desc C A-B CON	12,2,10,7 desc C A-B CON	12,20a, 2,14,16 desc C para CAWT Pasa a 8
	B contesta	NA	NA	NA	12,14,16 Pasa a 1
	Tono de CAW EXPIRADO	12,14,16 *marcar no *tono en A	NA	NA	12,14,16 * marcar no * tono en A

ESTADO DE LA LLAMADA					
		1. A-B hablando CAW tono to A. CI - I	2. A-C hablando B retenido CI - II	3. A-B hablando C retenido CI - I	4. A-B hablando CAW tono a A. B en TS. CI - I
A C C I O N E S	B DESC	12a,20a, 20a,3,10,7 A-C CON	12a,10,7 A-C CON	12a,3,10,7 A-C CON	12a,20a, 20a,3,10,7 A-C CON
	A DESC	12a,2,2,7 desc B,C	12a,2,2,7 desc B,C	12a,2,2,7 desc B,C	12a,2,2,7 desc B,C
D E L	C DESC	12a,20a, 10,7 A-B CON para CAWT	12a,3,10,7 A-B CON	12a,10,7 A-B CON	12a,20a, 14,16 A-B CON para CAWT ve a 8
	CIA1 C cuelga	NA	ve a C cuelga	Cambia B por C ve a 2	NA
S U B S C R I P T O R	CIA1	NA	16	NA	NA
	CIA1 cuelga env. tono	NA	14,16 ve a 12	NA	NA
	CIA2 on C	NA	5,20,14,16 ve a 13	NA	NA

A
C
C
I
O
N
E
S

D
E
L

S
U
B
S
C
R
I
P
T
O
R

ESTADO DE LA LLAMADA				
	5. A-B en hablandoC retenido B en TS. CI - I	6. A RB por C B desc. CI - III	7. A RB por C C desc. CI - IV	8. A-B en hablando B en TS. CI - V
A conmuta	12,4,2,3, 10,7 desc B A-C CON	NA	NA	12,16 Pasa a 8
A Cuelga	12,4,2, 20b,14,16 desc B Pasa a 6	NA	NA	12,4,2,2,7 desc A,B
B Cuelga	NA	NA	12,20a, 2,2,7 desc A,B para RB	NA
C Cuelga	12,2,14, 16 desc C Pasa a 8	12,20a, 2,2,7 desc A,C para RB	NA	NA
tono de llamada expira	NA	12,2,2,7 desc A,C	12,2,2,7 desc A,B	NA
A contesta	NA	12,3- 10,7 A-C CON	12,3, 10,7 A-B CON	NA
B contesta	12,14,16 Pasa a 3	NA	NA	12,10,7 A-B CON
B DESC	12,3,10, 7 A-C CON	NA	12a, 2,7 desc A	12,2,7 desc A
A DESC	12a,2,2,7 Desc B,C	12a,2,7 Desc C	12a,2,7 Desc B	12a,2,7 Desc B
C DESC	12a,14,16 Pasa a 8	12a,2,7 desc A para RB	NA	NA

		ESTADO DE LA LLAMADA		
		9 Conexión a A ha comenzado CI - VII	10 A llama C escucha tono CI - VII	11 A ocupado/ conges C ocupado tono CI - VII
A C C I O N	A contesta	NA	12,10,7 A-C CON	NA
H	C cuelga	12,2,7	12,2,7	12,2,7
D E L	C desc	12,7	12,7	12,7
S U B S C R I P T O R	Cong/ ocupado al esta blecer llamada	12,14,16 Pasa a 11	NA	NA
	Cone- xión de llamada	12,14,16 Pasa a 10	NA	NA

		ESTADO DE LA LLAMADA			
		12. A-C llamando B retenido retenido	13. A escucha ocupado B retenido	14. A-C llamando B desc.	15. A escucha ocupado B desc.
A C C I O N	A conmuta	4,2,3, 10,7 desc C A-B CON	6,3,10,7 A-B CON.	 NA	 NA
	C cuelga	16 Pasa a 2	 NA	10,7 A-C CON	 NA
D E L	B Cuelga	2,14,16 ve a 14	2,14,16 ve a 15	 NA	 NA
S U B S C R I P T O R	A Cuelga	4,2,20b, 14,16 Pasa a 7	6,20b, 16	4,2,2,7	6,2,7
	CIA1 on B	16	16	 NA	 NA
	Tono expirado	NA			6,2,7
	B DESC	14,16 ve a 6	14,16 ve a 15	 NA	 NA
	A DESC	2,2,7 Desc B,C	6,2,7 Desc B	2,7 Desc C	6,7
	C DESC	14,16 Pasa a 7	 NA	2,7 Desc A	 NA

ACCION O CONDICION RESULTANTE

IMAGEN DE LA LLAMADA

Imagen de la llamada:

I.	A -----+	II.	A -----+
	B -----+]		C -----+]
	C -----		B -----
III.	A -----	IV.	A -----
	C -----		B -----
V.	A -----+	VI.	A -----+
	B -----+]		C -----+]
VII.	C -----		

3.4 INSTRUCCIONES DE OPERACION**INTRODUCCION**

Estas instrucciones de operación abarcan:

- 1.- Conexión de LLAMADA EN ESPERA. Este procedimiento provee a un subscriptor específico el acceso a la facilidad de usuario LLAMADA EN ESPERA
- 2.- Modificación de LLAMADA EN ESPERA. Activación-Desactivación y la autorización al subscriptor para el uso de procedimientos del subscriptor para activar y desactivar la facilidad usando este procedimiento.
- 3.- Desconexión de LLAMADA EN ESPERA. Este procedimiento remueve de un subscriptor específico el acceso a la facilidad de usuario LLAMADA EN ESPERA

PREREQUISITOS

Una orden de trabajo deberá de ser presentada, conteniendo el número de directorio del subscriptor, y

además indicando si el subscriptor tendrá ó no la capacidad de activar/desactivar la facilidad por medio de un procedimiento de subscriptor. Esta orden deberá tener información acerca de si el subscriptor se encuentra en algún grupo especial, y si la facilidad será asignada a todo un grupo en especial ó solo a un individuo.

AYUDAS

Módulo E de documentación de la Central telefónica.

REFERENCIAS

Reporte de trabajo terminado.

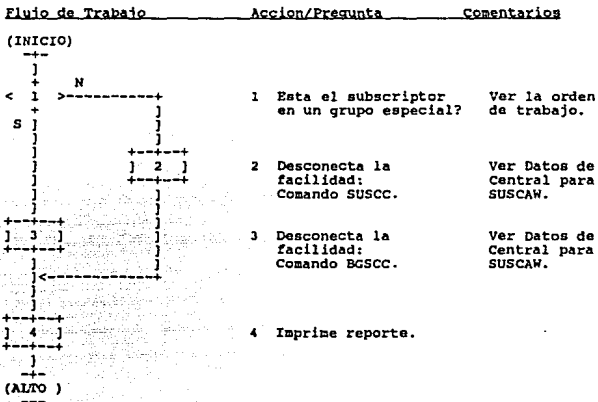
Descripciones de los comandos:

BGSCC	Cambio de clase en grupos especiales (Business Group, Subscriber Class, Change)
SUFIC	Cambio en la facilidad (Subscriber functions, Feature Information, Change)
SUSCC	Cambio en los datos del subscriptor (Subscriber data, Subscriber Class, Change)
SUFIP	Información sobre la facilidad (Subscriber functions, Feature Information, Print)
SUSCP	Información sobre la clase del subsc. (Subscriber data, Subscriber Class, Print)
SUSPP	Información de subscriptores especiales (Subscriber data, Subscriber with Specified Class, Print)

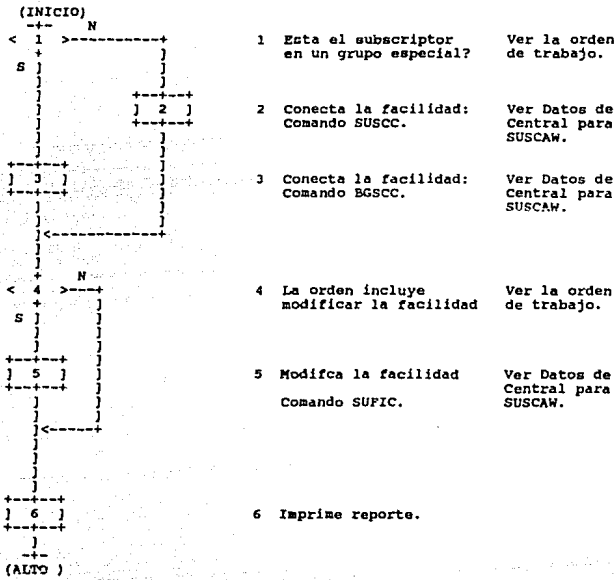
Datos de Central:

Bloque Funcional SUSCAW

DESCONECION DE LLAMADA EN ESPERA



CONEXION DE LLAMADA EN ESPERA

Flujo de Trabajo Accion/Pregunta Comentarios

MODIFICACION DE LLAMADA EN ESPERA

Flujo de Trabajo Accion/Pregunta Comentarios

(INICIO)

```

--+
  |
  |

```

```

--+--+
  |  |
  |  |

```

] 1]

```

--+--+
  |  |
  |  |

```

]]

]]

]]

```

--+--+
  |  |
  |  |

```

] 2]

```

--+--+
  |  |
  |  |

```

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

]]

1 Modifica la
 facilidad:
 Comando SUPIC.

Ver Datos de
 Central para
 SUSCAW.

2 Imprime reporte.

(ALTO)

```

--+
  |
  |

```

CAPITULO IV DISEÑO DEL BLOQUE FUNCIONAL**4.1 INFORMACION ACERCA DE LA REVISION DEL PRODUCTO**

La revisión de este producto se inicia debido a la necesidad de tener un software totalmente confiable y libre de problemas, por lo cual se requiere optimizar al máximo los recursos del sistema.

La revisión de este producto es iniciada apartir de lo especificado en los siguientes documentos :

- a) Especificación de los requerimientos para la facilidad de usuario LLAMADA EN ESPERA
- b) Propuesta de implementación para la facilidad de usuario LLAMADA EN ESPERA
- c) Limites de la función para LLAMADA EN ESPERA

RAZON DE LA REVISION

La facilidad de usuario es diseñada usando la filosofia del SISTEMA DE ADMINISTRACION DE TRAFICO (TMS), y como resultado de esto, se genera un nuevo bloque funcional llamado SUSCAW, el cual realizará ésta función.

BOLUCION TECNICA

Nuevo producto : Bloque funcional SUSCAW.

Programa fuente SUSCAW.

4.2 DESCRIPCION BLOQUE FUNCIONAL SUSCAW LLAMADA EN ESPERA**INTRODUCCION**

LLAMADA EN ESPERA es una facilidad de usuario cuya función es la de informar, por medio de un tono especial a un subscriptor que se encuentra ocupado, que otra llamada esta esperando a ser contestada.

IMPLEMENTACION

El bloque está totalmente implementado en la unidad central del procesador.

SEÑALES RECIBIDAS

CONTFS	incremento del archivo de usuario
COMMITR	acepta los cambios administrativos
CREATER	crea una ruta de un subscriptor a otro
CREATEFIELDR	crea un campo para un nuevo subscriptor
CREATETABLER	crea la base de datos
DELETEINFR	borra datos de una base de datos
DIDINFOACK	envia información estadística
EVENT	registra cualquier evento
FRER	libera cualquier conexión
GIVEFS	solicita incremento del archivo de usuario
HANDOVERR	pasa el control del tráfico a otra facilidad de usuario
INSERTINFR	inserta datos en la base de datos
JOINR	confirma unión entre usuarios
LOCRESOURCESR	reserva dispositivos para llamadas tripartitas
PREPARER	confirma los cambios administrativos
PROVIDEINSTRUC	dá instrucciones a SUSCAW
RECEIVEINFR	recibe información
RELCONTROLR	pasa el control del tráfico a otra facilidad de usuario
RELRESOURCESR	libera los recursos reservados
RETRIEVEINFR	trae información de la base de datos
ROLLBACKR	niega los cambios administrativos

SENDINFR	envia información
SENDRECEIVEINFR	envia/recibe información
SETFS	actualiza los archivos de usuario
SPLITR	separa cualquier conexión entre usuarios
STARTRANSACTR	verifica los cambios administrativos
STTOR	arranque del sistema
UPDATEINFR	actualiza las bases de datos

SEÑALES ENVIADAS

CONTFSEND	solicita cambios en el archivo del usuario
COMMIT	solicita cambios administrativos
CREATE	solicita una ruta entre dos abonados
CREATEFIELD	solicita espacio en la base de datos
CREATETABLE	solicita una base de datos
DELETEINF	solicita borrar datos de la base de datos
DIDINFO	solicita datos estadísticos
FINISHED	termina SUSCAW
FREE	libera conexiones
GIVEFSEND	establece archivo de usuario
HANDOVER	solicita otra facilidad de usuario
INSERTINF	solicita insertar información en la base de datos
JOIN	require unir dos abonados
LOCRESOURCES	requisición de recursos
MONITOREVENT	monitorea los eventos
PREPARE	prepara cambios administrativos
READYFOREVENT	espera eventos

RECEIVEINF	solicita información
RELCONTROL	solicita liberar el control de la llamada
RELRESOURCES	solicita liberar los recursos reservados
RETRIEVEINF	solicita información administrativa
ROLLBACK	solicita negar cambios administrativos
SENDINF	solicita enviar información
SENDRECEIVEINF	solicita/envia información
SETFSEND	solicita cambios de archivo de usuario
SPLIT	solicita unir dos abonados
STARTTRANSACT	inicia cambios administrativos
STORRY	arranque del sistema
UPDATEINF	actualiza bases de datos

SEÑALES INTERNAS CP-CP

CONTINUEC	Continúa procesamiento en nivel C
-----------	-----------------------------------

FUNCION**INTRODUCCION**

SUSCAW, maneja LLAMADA EN ESPERA (CAW) y LLAMADA EN ESPERA ENTRANTE (CWI) dos facilidades de usuario que trabajan de la misma forma solo que cuando CWI esta en uso ninguna llamada que provenga desde la red pública es aceptada.

El único bloque funcional que mantiene una relación directa con SUSCAW es FSH el cual realiza las tareas ordenadas por SUSCAW, tareas como el envío de información o la extracción de datos de las tablas del sistema.

Las siguientes definiciones son usadas en este documento:

Esta es la imagen inicial de LLAMADA EN ESPERA, obviamente ésta imagen irá cambiando de acuerdo a las acciones de los subscriptores

```
A ----+
      | <-- llamada normal entre 2 usuarios
B ----+
C ----- <-- subscriptor intruso
```

Definiciones.

- A Subscriptor con la facilidad de LLAMADA EN ESPERA
- B Subscriptor cualquiera, en conversación con A
- C Intruso en la llamada A-B u originador de la llamada en espera

SUBFUNCIONES NORMALES

CARGA INICIAL Y RECARGA DEL SISTEMA

Cuando el sistema es puesto en funcionamiento por primera vez (arranque) ó bien cuando es puesto en funcionamiento nuevamente después de algún paro (rearranque) SUSCAW recibe la señal STTOR para saber en que fases de la carga del sistema, SUSCAW actuara, SUSCAW entonces hace lo siguiente dependiendo de la fase en que se encuentre :

Fase 1 .- Fase absoluta 21, SUSCAW carga la referencia del bloque

Fase 2.- Fase absoluta 24, SUSCAW envia las señales FEATURETEXT con los datos de LLAMADA EN ESPERA , y también durante ésta fase crea las tablas donde cada usuario que tiene ésta facilidad es registrado.

Fase 4.- Fase absoluta 29, hace un rastreo para saber que llamadas fueron cortadas y cuales no.

Fase 7.- Fase absoluta 44, SUSCAW envia la señal DIDINFO con datos de contadores para fines estadísticos.

RASTREO DE LA LLAMADA

Cuando el sistema es interrumpido por algún problema, se inicia una rutina de mantenimiento para saber cual es la

causa de la interrupción, en este momento, SUSCAW también inicia un rastreo de las llamadas, para saber el estado de la llamada de cada usuario que estaba hablando, si el usuario estaba hablando, la llamada es recuperada de tal manera que ambos usuarios podran seguir hablando sin percibir esta interrupción garantizando con ello un eficaz servicio, si los usuarios no estaban en linea, es decir todavia no se establecia la comunicación entre usuarios, el sistema simplemente los desconecta y reconecta nuevamente.

CAPACIDAD DE MANEJO DE USUARIOS

SUSCAW también tiene la facilidad o habilidad de manipular el tamaño de archivo de usuarios, es decir puede añadir ó quitar tantos subscriptores como el requiera, esto lo hace por medio de las señales GIVEFS, GIVESFSEND, CONFES, CONFESSEND, SETSF, SETSFSEND y también por medio de CONTINUEC, usando estas señales, SUSCAW logra el manejo de "listas de individuos" es decir los archivos donde están almacenados la información de cada usuario, verificando, analizando y si es permitido, incrementando ó reduciendo éste archivo segun sus necesidades, el limite minino del archivo es 0 y el máximo 65535.

SUBFUNCIONES DE APLICACION TOMA DE UN INDIVIDUO

Cuando la señal ProvideInstruc es recibida con un código de administración ó tráfico, SUSCAW lo primero que hace es localizar dentro de la "lista de individuos libres" de su archivo para ver si puede atender esa solicitud, si no existe ningún recurso disponible, el usuario recibirá un tono de ocupado, si el recurso es encontrado, SUSCAW continua con el manejo de los abonados.

ADMINISTRACION DE LA BASE DE DATOS

PROVISION DE LLAMADA EN ESPERA

Cuando LLAMADA EN ESPERA es asignada a un subscriptor, SUSCAW es informado por medio de la señal ProvideInstruc, entonces SUSCAW, realiza las siguientes acciones:

Lee el apuntador de usuario por medio de las señales ReceiveInf/R.

Prepara la base de datos para cambios administrativos por medio de las señales Starttransact/R .

Inserta los datos del nuevo usuario en la base de datos por medio de las señales InserInf/R.

Actualiza todas las bases de datos del sistema por medio de UpdateInf/R.

Hace la confirmación de todos estos cambios por medio de Prepare/R y Commit/R.

Actualiza los contadores del sistema por medio de Sendinf/R.

Finalmente libera la conexión del subscriptor por medio de Free/R y termina la ejecución de programa por medio de Finished.

CANCELACION DE LLAMADA EN ESPERA

Cuando LLAMADA EN ESPERA es cancelada a un subscriptor, SUSCAW es informado por medio de la señal ProvideInstruc, entonces SUSCAW realiza las siguientes acciones:

Lee el apuntador de usuario por medio de las señales ReceiveInf/R

Prepara la base de datos para cambios administrativos por medio de las señales Starttransact/R

Cierra permanentemente las bases de datos por medio de RetrieveInf/R para así evitar que la información contenida en ese momento sea cambiada.

Verifica por medio de estas mismas señales que el usuario en cuestión éste realmente definido en la base de datos, y si tiene algún usuario de Terminal 1 definido, si el usuario esta activo, SUSACW lo desactiva. Si el usuario tiene asignado un subscriptor Terminal 1, éste Terminal 1 es checado.

Borra los datos del usuario en la base de datos por medio de las señales DeleteInf/R.

Actualiza todas las bases de datos del sistema por medio de UpdateInf/R.

Hace la confirmación de todos estos cambios por medio de Prepare/R y Commit/R.

Actualiza los contadores del sistema por medio de Sendinf/R

Y finalmente libera la conexión del subscriptor por medio de Free/R y termina la ejecución de programa por medio de Finished.

CAMBIO EN LA FACILIDAD POR MEDIO DE COMANDO

Cuando LLAMADA EN ESPERA es cancelada a un suscriptor, SUSCAW es informado por medio de la señal ProvideInstruc, esto es el comando SUFIC ha sido dado en la central telefónica para modificar los parámetros del servicio del usuario entonces SUSCAW realiza las siguientes acciones:

Por medio de las señales ReciveInf/R recibe los nuevos datos del suscriptor.

Prepara la base de datos para cambios administrativos por medio de las señales Starttransact/R.

Por medio de las señales RetrieveInf/R la base de datos es cerrada temporalmente para evitar cambios en la información almacenada, además se obtienen los datos que ya estaban almacenados con en la base de datos, con la finalidad de saber cuales serán los cambios a realizar.

Los datos son analizados por SUSCAW, de tal manera que si los cambios son aceptables, entonces las bases de datos son actualizadas por medio de UpdateInf/R.

Hace la confirmación de todos estos cambios por medio de Prepare/R y Commit/R.

Actualiza los contadores del sistema por medio de Sendinf/R y finalmente libera la conexión del suscriptor por medio de Free/R y termina la ejecución de programa por medio de Finished.

MANEJO DE ERRORES EN LA ADMINISTRACION

Si durante la provisión, Cancelación ó Cambio de parámetros de la facilidad de usuario ó administración de LLAMADA EN ESPERA, algo malo ocurre ó algún dato es erróneo, el procedimiento a ejecutar es rechazado, se envía un código de error por medio de las señales Sendinf/R, se cancela cualquier acción en las bases de datos por medio de Rolback/R , actualiza los contadores del sistema por medio de Sendinf/R y finalmente libera la conexión del suscriptor por medio de Free/R y termina la ejecución de programa por medio de Finished.

PROCEDIMIENTOS DE SUBSCRIPTOR

SUSCAW es informado por medio de la señal ProvideInstruc, que algun usuario ha usado el código que puede activar ó desactivar la facilidad de usuario.

Cuando esto ocurre, SUSCAW lee el apuntador de ese subscriptor por medio de la señal ReceiveInf/R, éste dato lo obtiene leyendo en las tablas de datos de LLAMADA EN ESPERA de estas tablas también se obtiene el contador de subscriptores Terminal 1, dato con el cual SUSCAW podrá determinar si el intento de activar ó desactivar la facilidad de usuario fue realizado por el propio subscriptor ó fue hecho por su Terminal 1.

De acuerdo a lo anterior SUSCAW tiene dos procedimientos de usuario que manejar, uno el el Procedimiento de subscriptor realizado por el propio subscriptor, y el otro es el Procedimiento de subscriptor realizado por su terminal 1.

PROCEDIMIENTO DE SUBSCRIPTOR REALIZADO POR EL PROPIO SUBSCRIPTOR

Cuando SUSCAW ha confirmado que es el propio subscriptor es el que ha marcado el código de activar/desactivar, SUSCAW analiza el parámetro de la Información del subscriptor SCI por medio del cual sabe si el subscriptor esta habilitado ó no para realizar este procedimiento, éste parámetro se verifica de la siguiente manera :

- a) Si el valor del parámetro SCI es igual a NONE, esto significa que el subscriptor no esta habilitado para éste procedimiento y por lo tanto, por medio de las señales SENDINF/R, se envia el código de error, lo cual se traduce al subscriptor en un tono de reorden.
- b) Si el valor del parámetro SCI es igual a ACT, esto significa que el subscriptor tiene la habilidad de activar y desactivar la facilidad desde su aparato telefónico.

Cuando el subscriptor si esta habilitado para activar/desactivar la facilidad de usuario, por medio de las señales SendInf/R, Prepare/R y Commit/R las bases de datos son actualizadas, también el tono de confirmación es enviado al subscriptor, y después de que este tono termina ó que el subscriptor cuelga su aparato telefónico, es liberada la conexión y así mismo la facilidad de usuario y también las listas enlazadas de

usuarios son actualizadas, así como los contadores de subscriptores activos e inactivos también son actualizados.

PROCEDIMIENTO DE SUBSCRIPTOR POR MEDIO DEL SUBSCRIPTOR TERMINAL 1

Una vez que SUSCAW determinó que el procedimiento es realizado por medio de un subscritor terminal 1, manda un tono de marcación, en este momento el subscritor marca el código de activación, y estos dígitos son recogidos por SUSCAW y enviados a analizar por medio de la señal SendReceiveInf/R, una vez que estos dígitos son analizados, la respuesta de si son o no correctos, es recibida en la señal Event.

Cuando los dígitos son correctos entonces SUSCAW realiza el siguiente procedimiento para verificar el SC11 y así determinar si éste subscritor está habilitado para realizar este procedimiento:

- a) Si el valor del parámetro SC11 es igual a NONE, esto significa que el subscritor no está habilitado para este procedimiento y por lo tanto, por medio de las señales SENDINF/R, se envía el código de error lo cual se traducirá al subscritor como un tono de reorden.
- b) Si el valor del parámetro SC11 es igual a ACT, esto significa que el subscritor tiene la habilidad de activar y desactivar la facilidad desde su aparato telefónico.

Cuando el subscritor si está habilitado para activar/desactivar la facilidad de usuario, por medio de las señales SendInf/R, Prepare/R y Commit/R las bases de datos son actualizadas, también el tono de confirmación es enviado al subscritor, y después de que este tono termina ó que el subscritor cuelga su aparato telefónico, es liberada la conexión y así mismo la facilidad de usuario y también las listas enlazadas de usuarios son actualizadas, así como los contadores de subscriptores activos e inactivos también son actualizados incluyendo también al contador de subscriptores tipo Terminal 1.

MANEJO DE ERRORES DURANTE LOS PROCEDIMIENTOS DE SUBSCRIPTOR

Cuando algún error es detectado durante estos procedimientos de subscritor, alguna señal regresa con un código de error ó el subscritor hace algún

procedimiento invalido, SUSCAW por medio de las señales Rollback/R y SendInf/R envia el código de error al sistema, lo que al subscriber se traduce en algún mensaje de error ó un tono de reordenamiento.

Cuando éste tono termina ó que el subscriber cuelga su aparato telefónico, es liberada la conexión y así mismo la facilidad de usuario y también las listas enlazadas de usuarios son actualizadas, así como los contadores de subscribers activos e inactivos también son actualizados incluyendo también al contador de subscribers tipo Terminal 1.

ESTABLECIMIENTO DE LA LLAMADA (MANEJO DE TRAFICO)

Cuando algún subscriber con LLAMADA EN ESPERA, esta ocupado en una conversación telefónica, y en ese momento algún otro usuario trata de hablar con el, LLAMADA EN ESPERA entra en funcionamiento.

SUSCAW recibe la señal ProvideInstruc con la identidad del subscriber C iniciando funcionamiento del bloque, así SUSCAW por medio de las señales ReceiveInf/R consigue la identidad del subscriber A y del subscriber B.

Por medio de las señales SendInf/R envia la información de que ese usuario esta haciendo uso de la facilidad de usuario, con lo cual se asegura que SUSCAW siempre manejará una y sólo una llamada en espera al mismo tiempo para cada subscriber, de tal manera que las demás llamadas en espera recibirán un tratamiento de como si el suscriptor estuviese ocupado.

SUSCAW ahora por medio de las señales Create/R crea una ruta para conectar los subscribers A y C, así cuando la ruta ha sido creada, se envia la información (por medio de las señales SendInf/R) de que ahora el subscriber C será el que controle la llamada, no importando quien fue el que la inició.

Ahora SUSCAW analiza el tipo de facilidad de usuario que el subscriber tiene asignada CAW ó CWI y también determina la ubicación de los usuarios para así determinar si la llamada es ó no permitida, esto es realizado por medio de las señales ReceiveInf/R/.

Cuando la facilidad de usuario requerida es LLAMADA EN ESPERA ENTRANTE, si los dos subscribers se encuentran dentro del mismo grupo de distribución de la central telefónica, la llamada no es aceptada y el subscriber recibirá un tono de ocupado y la facilidad de usuario así como las conexiones son liberadas.

Quando la llamada es permitida, SUSCAW verifica si el subscriptor tiene la facilidad de usuario LLAMADA RETENIDA, si es así, SUSCAW envía un tono para que el usuario marque el código de retención de llamada y así SUSCAW transfiere el control de la llamada al bloque funcional que contiene la facilidad de usuario LLAMADA RETENIDA, esto es realizado por medio de las señales Handover/R, finalmente SUSCAW es liberado.

Si el subscriptor no tiene la facilidad de LLAMADA RETENIDA, entonces SUSCAW será el encargado de manejar el tráfico. Para esto lo primero que SUSCAW hace es reservar circuitos para manejo de conexiones tripartitas por medio de las señales LocResources/R y una vez reservados estos dispositivos, entonces SUSCAW por medio de las señales SendInf/R envía al subscriptor A el tono de LLAMADA EN ESPERA y al subscriptor C el tono llamada.

Una vez que SUSCAW ha enviado los tonos, SUSCAW por medio de las señales MonitorEvent y Event indica al sistema que esta listo para recibir la información de los subscriptores es decir SUSCAW espera la información de que acción tomo el subscriptor (colgo, se desconectó, contestó la llamada en espera, no la contestó, etc.) y así de esta manera saber como responder a ésta acción, como se encuentra estipulado en la DESCRIPCION DE LA FUNCION en la parte de las tablas de estados.

MANEJO DE LOS ERRORES DURANTE EL MANEJO DE TRAFICO

Quando alguna señal regresa con algún código de error, el subscriptor realiza alguna maniobra no permitida ó algún error es detectado, SUSCAW libera las conexiones que no estén en posición de habla y posteriormente liberara a la facilidad de usuario así como todas las conexiones asociadas.

4.3 DATOS DE CENTRAL DE LLAMADA EN ESPERA BLOQUE FUNCIONAL SUSCAW

FUNCION

LLAMADA EN ESPERA (CW) es una facilidad de usuario que informa a un subscriptor que una tiene una llamada

esperando a ser contestada en la misma línea telefónica que el está usando en ese momento.

TAMAÑO DE ARCHIVO

Un individuo dentro del archivo de SUSCAW puede ser utilizado ó tomado, cuando se dá un comando, cuando se está en un procedimiento de subscriptor ó bien durante el manejo de tráfico, este individuo es liberado cuando todas las tareas encomendadas a LLAMADA EN ESPERA han sido terminadas

Comando Parámetro	Rango de valor	Descripción
SAE	500	Archivo que será alterado
NI	0-65535	Número de individuos longitud de incremento 1 individuo.
BLOQUE	1C-7C	Designación del bloque funcional que deberá de ser indicado cuando se lleve a cabo la alteración del archivo de usuario, Este valor esta definido por el sistema y es : SUSCAW.

DATOS DE TRAFICO

PROCEDIMIENTOS DE SUBSCRIPTOR PARA ACTIVACION/DESACTIVACION DE LLAMADA EN ESPERA

El subscriptor puede ser ó no provisto con la facilidad de habilitar o deshabilitar la facilidad de usuario LLAMADA EN ESPERA, por medio de algún procedimiento de subscriptor, Cuando la facilidad esta desactivada, ninguna llamada en espera es permitida, y ésta facilidad permanecerá desactivada hasta que sea activada por algún comando del operador en la central telefónica ó por algún procedimiento del propio subscriptor en su aparato telefónico.

Un subscriptor con CAW/CWI podrá tener otro individuo capaz de realizar los procedimientos de usuarios por el, este individuo es llamado SUBSCRIPTOR TERMINAL 1 y para que un subscriptor tenga un Terminal 1, en la central telefónica deberá de darse el comando SUFIC con el parámetro SNB1.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

Dos "códigos de servicios" son normalmente establecidos durante la preparación de los datos de central, uno es usado por el cliente para avizar a la central telefónica que desea activar la facilidad de usuario CAW/CWI, y el otro es usado por el cliente para avizar a la central telefónica que desea desactivar la facilidad de usuario CAW/CWI.

Para realizar un análisis más simple de los datos y una respuesta más rápida al subscriptor, los códigos de servicio usados como prefijos (* ó 11) son definidos junto con los datos de Análisis de datos de Central.

También y para mayor comodidad del usuario existe la opción de usar como códigos de servicio, sufijos (ejemplo # ó también la expiración del tiempo).

Sin embargo la determinación de estos prefijos y sufijos es dependiente del mercado en que se encuentre ésta facilidad de usuario.

Los siguientes procedimientos son usados por el subscriptor para activar ó desactivar la facilidades de usuario de llamada en espera.

Notación usada:

CT = Tono de confirmación.

SCA = Código de Servicio para activación de LLAMADA EN ESPERA.

SCD = Código de Servicio para desactivación de LLAMADA EN ESPERA.

SCP = Código de Servicio prefijo (* ó 11).

SCS = Código de Servicio sufijo (# ó expiración de tiempo).

PROCEDIMIENTO NORMAL DE ACTIVACION

(SCP) SCA CT

(SCP) SCA RDT SNB CT (Subscriber Terminal 1)

Este procedimiento usa un prefijo seguido por el código de activación de servicio, el programa de envío en éste caso esta definido por en número de digitos en el SCP y el SCA, el primer dígito para la transmisión de digitos es el siguiente dígito.

Ejemplo 1

*38 CT

Aquí un teléfono con botonera es usado; el SCP es "*" y el SCA es "38". Así, el programa de envío es 334.

Ejemplo 2

1138 CT

Aquí un teléfono de teclado ó uno de disco puede ser usado, el prefijo es "11" y el código de servicio es "38". Por lo tanto el programa de envío es 445.

Ejemplo 3

Por medio de un Subscriber Terminal 1

*38 RDT SNB CT

Aquí un teléfono con botonera es usado; el SCP es "*" y el SCA es "38". Así, el programa de envío es 334.

Comando Parámetro	Rango de valor	Descripción
R	1C-7C	Nombre global de la ruta o ruta de salida.
SP		Programa de envío para caso de enrutamiento.
DETY	1C-7C	Tipo de dispositivo para dispositivos usados con ruta Valor recomendado = FSOIA.
FNC	138	Código para activación de CAW.
FNC	298	Código para activación de CWI.

PROCEDIMIENTO NORMAL DE DESACTIVACION

(SCP) SCD CT

(SCP) SCD RDT SNB CT (Subscriber Terminal 1)

Este procedimiento usa un prefijo seguido por el código de desactivación de servicio, el programa de envío en éste caso esta definido por en número de dígitos en el SCP y el SCA, el primer dígito para la transmisión de dígitos es el siguiente dígito.

Ejemplo 1

*38 CT

Aquí un teléfono con botonera es usado; el SCP es "*" y el SCD es "38". Así, el programa de envío es 334.

Ejemplo 2

1138 CT

Aquí un teléfono de teclado ó uno de disco puede ser usado; el prefijo es "11" y el código de servicio es "38". Por lo tanto, el programa de envío es 445.

Ejemplo 3

Por Subscriber Terminal 1

*38 RDT SNB CT

Aquí un teléfono con botonera es usado; the SCP es "*" y el SCD es "38". Así, el programa de envío es 334.

Comando	Rango de valor	Descripción
R	1C-7C	Nombre global de la ruta o ruta de salida.
SP		Programa de envío para caso de enrutamiento.
DETY	1C-7C	Tipo de dispositivo para dispositivos usados con ruta Valor recomendado = FSOIA.
FNC	139	Código para desactivación de CAW.
FNC	299	Código para desactivación de CWI.

MANEJO NORMAL DE TRAFICO EN BUSCAW

Comando Parámetro	Rango de valor	Descripción
R	1C-7C	Nombre global de la ruta o ruta de salida.
DETY	1C-7C	Tipo de dispositivo para dispositivos usados con ruta Valor recomendado = FSOIA.
FNC	137	Código para manejo normal de tráfico en CAW.
FNC	297	Código para manejo normal de tráfico en CWI.

INTERACCION ENTRE LLAMADA EN ESPERA Y LLAMADA RETENIDA

Comando Parámetro	Rango de valor	Descripción
R	1C-7C	Nombre global de la ruta o ruta de salida.
DETY	1C-7C	Tipo de dispositivo para dispositivos usados con ruta Valor recomendado = FSOIA.
FNC	140	Código para manejo normal de interacción HLD-CAW.
FNC	300	Código para manejo normal de interacción HLD-CWI.

CODIGOS DE SELECCION DE TERMINO DE LLAMADA

Parámetro	Rango de valor	Descripción
ES	2221	El código de servicio ha sido Activado: Tono de confirmación.
ES	2222	El código de servicio ha sido desactivado: Tono de confirmación.

ES	2223	Congestión: Tono de congestión
ES	1457	Intento realizado por un inválido terminal 1: Tono de volver a marcar.
ES	2236	SCI no permite el uso del servicio: Tono de volver a marcar.
ES	2237	El subscriptor no tiene la facilidad asignada Tono de volver a marcar.

DATOS DEL SUBSCRIPTOR

Comando

Parámetro	Rango de valor	Descripción
SCL	CAW-0 CAW-1	Cancelación CAW. Asignación CAW.
	CWI-0 CWI-1	Cancelación CWI. Asignación CWI.
FEATURE	CAW	Abreviación de la facilidad de usuario.
	CWI	Abreviación de la facilidad de usuario.
ACT		Activación de LLAMADA EN ESPERA.
DEACT		Desactivación de LLAMADA EN ESPERA.
SNB	snb	Número de directorio del subscriptor.
SCI	ACT	Subscriptor puede cambiar su estado de activación.
	NONE	Subscriptor no tiene acceso al procedimiento de subscriptor.
	STD	Acceso normal a cualquier procedimiento.
SNB1	snb1	Número de directorio del subscriptor Terminal 1.
SCI1	ACT	Subscriptor Terminal 1 puede cambiar su estado de activación.

NONE	Subscriber Terminal 1 no tiene acceso al procedimiento de subscriber.
STD	Terminal 1 acceso normal a cualquier procedimiento.

DATOS DE GRUPO DE DISTRIBUCION

Comando

Parámetro	Rango de valor	Descripción
-----	-----	-----

SCL	CWI-0 CWI-1	Cancelación CWI. Asignación CWI.
BG	bg	Designación de grupo de distribución.
BGS	bgs	Número de grupo de distribución.
BSNB	bsnb	Número de subscriber en el grupo de distribución.

CAPITULO 5

CONCLUSIONES

La tendencia mundial en materia de comunicaciones esta dirigida hacia su automatización total. En el aspecto telefónico, destaca la importación de este hecho, ya que en la medida en que el sistema telefónico nacional evolucione se podrá observar un mejor desarrollo del país.

La automatización de un sistema implica principalmente el uso de computadoras para controlar un proceso, en el caso de un sistema telefónico es de vital importancia que al automatizarse el sistema el controlador del mismo (computadora o procesador central) este perfectamente programado para brindar un servicio de alta calidad y evitar cualquier tipo de problema; problemas que irían por ejemplo desde cobros indebidos, hasta la total suspensión del servicio.

Desde el punto de vista de mantenimiento y operación del sistema, cuando algún equipo telefónico automatizado falla, el primer paso para corregir esa falla es tratar de localizarla.

La mayoría de las veces esto es manejado como falla de algún elemento físico -o falla de hardware- (ruptura de cables, sobrecalentamiento de equipo, falta de energía en partes del sistema, etc.) y lamentablemente el último paso que se dá al respecto es el revisar que los programas que controlan el sistema estén trabajando correctamente -falla en el software-.

La confiabilidad actual de los sistemas es del 99.9 % y para ello aún en el caso de falla existe redundancia de partes principales.

Es aquí donde radica la importancia de que el procesador central de la central telefónica digital tipo SPC cuente con un sistema de programación de excelente calidad, bien estructurado, capaz de reaccionar ante cualquier circunstancia, y situaciones inesperadas.

Para muestra baste un botón:

Enero 15 de 1990, la red telefónica más grande del mundo, y a su vez la más avanzada tecnológicamente, fue en totalmente paralizada, la causa?? una falla en la programación del sistema, si un pequeñísimo error (insignificante??) en la programación del sistema logro la paralización completa de la red por un espacio de 9

horas. Este problema comenzó a las 14:25 hrs y pudo ser controlado hasta las 23:30 hrs. del mismo día.

Resultado, la totalidad de las 114 centrales telefónicas tipo 4ESS de AT&T fueron totalmente paralizadas, esto significó la suspensión total de llamadas de larga distancia, larga distancia internacional, servicio 800, algo así como 40 millones de llamadas telefónicas, debido esta "pequeña" falla, de un programa del procesador central, sin tomar en cuenta las pérdidas económicas y el prestigio perdido.

El problema descrito anteriormente, ha sido el más grande ocurrido en la historia de la telefonía, no obstante es frecuente leer en revistas ó diarios extranjeros acerca de paros de redes telefónicas, corrupción de datos, etc. Es verdad que éste problema no se dió en México, pero la pregunta es :

¿ Debemos esperar a que ocurra en México algo similar para tomar cartas en el asunto ?

LA RESPUESTA OBIAMENTE ES NO

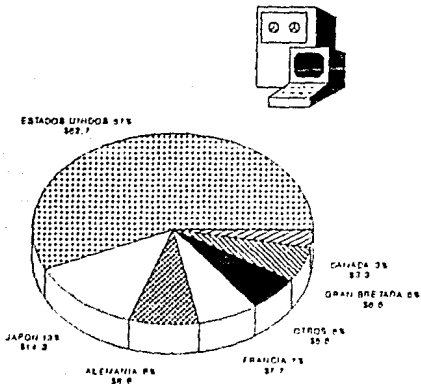
Ahora bien podrá observar de la gráfica de la página siguiente, México abarca una pequeñísima parte dentro del mercado mundial del software, lo cual trae como consecuencia una gran dependencia del extranjero. Siendo éste un mercado importantísimo tanto política como tecnológicamente es altamente recomendable iniciar acciones que nos lleven a figurar en éste mercado, aunque la verdad la tarea no sea nada fácil tomando en cuenta quienes son los actuales líderes de éste mercado.

El diseño de la facilidad de usuario "LLAMADA EN ESPERA" aquí presentado fué exitosamente compilado, cargado y probado en las centrales de prueba AXE 10 de ERICSSON, también la documentación correspondiente al diseño del sistema y al diseño del bloque ha sido minuciosamente inspeccionada, siguiendo los requisitos que para ello se marcan los estadarcs de calidad ISO-9000. Cabe señalar que fué la base para un proyecto de diseño de software instalado de enero a marzo de 1991 en WATERTOWN ciudad cercana a New York, lo cual es indicativo de la alta calidad del producto.

Debido a lo anterior se hace urgente la necesidad de que al mismo tiempo que la red se moderniza con equipo altamente tecnológico, también se cuente con apoyo para los sistemas de programación de excelente calidad, así como también una plantilla de personal MEXICANO altamente calificado capaz de resolver cualquier contingencia y que no solo se dedique a recibir y estudiar tecnología

extranjera, si no que también diseñe e implemente y desarrolle sistemas nacionales, mostrando al mundo entero que la ingeniería en México está al nivel de cualquier otra en el mundo, evitando así la gran dependencia tecnológica que actualmente sufrimos.

MERCADO MUNDIAL DE SOFTWARE



BILLONES DE DOLARES INVERTIDOS

JOSE MARIO MIRANDA MARTINEZ
SEPTIEMBRE 1991

GLOSARIO

1C-7C	Designación del bloque funcional de 1 a 7 caracteres
A	Subscriber ó abonado que tiene asignada al facilidad de usuario de "LLAMADA EN ESPERA"
A-B	Llamada normal entre el abonado A y el abonado B
A-C	Llamada normal entre el abonado A y el abonado C
ABJ	A-B Junctor dispositivo usado para el establecimiento de una llamada normal entre dos usuarios
ACT	Activación de LLAMADA EN ESPERA
B	Subscriber ó abonado normal
BG	Designación de grupo de distribución (business group)
BGS	Número de grupo de distribución (business group)
BGSCC	Comando de Cambio de clase en grupos especiales
BS	Grupo Especial en la central telefónica Business group
BSNB	Número de subscriber
C	Subscriber normal, que intenta comunicarse con subscriber A cuando éste esta ocupado.
CAW	LLAMADA EN ESPERA (CAW Call Waiting)
CAW-1	Categoría de subscriber con LLAMADA EN ESPERA
CAWT	Tono de llamada en espera
CI	Imagen de la llamada
CIA	Imagen de la llamada ha sido alterada

COD	Decodificadores de direcciones
CON	Conexión normal entre dos abonados
CP	Procesador Central
CPU	Unidad Central de Proceso
CT	Tono de confirmación
CWI-1	Categoría de suscriptor con LLAMADA EN ESPERA ENTRANTE
CWI	LLAMADA EN ESPERA ENTRANTE
D C	Corriente Directa
DAS	Almacen de Datos (Data Store)
DEA	Registro de direcciones
DEACT	Desactivación de LLAMADA EN ESPERA
desc	Desconexión de la llamada
DETY	Tipo de dispositivo
DTMF	Teléfono de multifrecuencias
ES	El código de servicio
FDM	Multiplexaje por División de Frecuencia (Frequency División Multiplex).
FEATURE	Facilidad de usuario
FLA	Commutación realizada por el suscriptor
FNC	Código de función
FSH	Bloque Funcional que controla el uso de las facilidades de usuario (feature support Handling)
FSOIA	Bloque funcional
HLD-CAW	Interacción entre LLAMADA EN ESPERA y LLAMADA EN RETENCION
HLD-CWI	Interacción entre LLAMADA EN ESPERA y LLAMADA EN RETENCION
KR	Receptor de código marcado (key receiver)
LAN's	Red local (local area network)

LI	Interface de linea (line interface)
MLHG	Equipo multilinea (multi line hunt group)
MMS	Subsistema de comunicación hombre-máquina (Man Machine Subsystem)
MTBF	Tiempo promedio entre falla (mean time between fault)
NA	Combinación No Aceptada en el uso de Llamada en Espera
NI	Número de individuos 0-65535
NONE	Subscriber no tiene acceso a realizar procedimientos de subscriber
OIS	Subsistema de interface entrada-salida (output-input machine subsystem)
PBX	Central telefónica privada
PC	Computadora personal
PCM	Multiplexión por división de tiempo (Pulse Code Modulation)
PLEX	Lenguaje de Programación de Alto nivel para centrales telefónicas (Program Language for exchanges)
PRS	Almacen de Programas (PProgram Store)
R	Nombre global de la ruta entre dos abonados
RB	Retrollamada al subscriber
RDSI	Red Digital de servicios Integrados
RDT	Tono de retrollamada
REO	Orden de lectura
RWS	Leer palabra del almacen (read word store)
SAE	Archivo de usuario que será alterado (size alteration)
SC	Categoría del Subscriber
SCA	Código de Servicio para activación de LLAMADA EN ESPERA

SCA	Código de servicio para activación de la LLAMADA EN ESPERA
SCCP	Nodo de control y servicio de una red con señalización por canal común
SCD	Código de Servicio para desactivación de LLAMADA EN ESPERA
SCH	Código de servicio para la facilidad de LLAMADA EN ESPERA
SCI	Parámetro de Control de la información acerca del Subcriptor con LLAMADA EN ESPERA
SCI1	Parámetro de Control de la información acerca del Subcriptor terminal 1
SCL	Clase de subcriptor
SCP	Código de Servicio prefijo (* ó 11)
SCS	Código de Servicio sufijo (# ó expiración de tiempo)
SFSHD	Bloque funcional para la función de LLAMADA EN RETENCION
SNB	Número de directorio del subcriptor
SNB1	Número de directorio del subcriptor terminal 1
SNR	Juego de relevadores del circuito de cordón
SP	Programa de envío
SPC	Control por programa almacenado
SS7	Señalización de canal común
TS	Tiempo de supervisión
STD	Subcriptor tiene/no tiene acceso a realizar procedimientos de subcriptor parámetro dependiente de la aplicación
STP	Nodo de señalización una red con señalización por canal común
SUFIC	Comando para Cambios en la facilidad
SUFIP	Comando para Información sobre la facilidad

SUS	Subsistema del subscriptor
SUSCAW	Bloque funcional que aloja la función LLAMADA EN ESPERA
SUSCC	Comando para Cambio en los datos del subscriptor
SUSCP	Comando para Información sobre la clase del subscriptor
SUSPP	Comando para Información de subscriptores especiales
TANDEM	Tipo de central telefónica
TCS	Subsistema del control de tráfico
TELMEX	Teléfonos de México
TERMINAL 1	Subscriptor que tiene la facilidad de activar o desactivar un afacilidad de usuario de otro subscriptor
TMS	Sistema de administración de tráfico
TO	Tiempo de supervisión expirado
TS	Tiempo de supervisión
WRO	Orden de escritura

BIBLIOGRAFIA

- 1.- GETTING TO KNOW AXE
EDITADO POR ERICSSON STOCKHOLM 1986
- 2.- BREVE INTRODUCCION A LAS TELECOMUNICACIONES
EDITADO POR TIM/CELE ERICSSON MEXICO 1985
- 3.-METODOS
AUT. JEREZ-GRIJALVA ED. LIMUSA
- 4.-REFERENCIAS TECNICAS DE ESPECIFICACIONES DE BELL-CORE
LBSGR-TR-TSY-000064 "CALL WAITING" USA
- 5.-LIBRO AZUL TOMO III FASCICULO III.7 DEL CCITT
MELBOURNE NOVIEMBRE 1988
- 6.-REFERENCIAS TECNICAS DE ESPECIFICACIONES DE ALCATEL-
INDETEL "LLAMADA EN ESPERA"
- 7.-MANUAL DE OPERACION "SYSTEM OVERVIEW" GTE-OMNI 1000
EDITADO POR GTE 1980
- 8.-INTRODUCCION A LA TECNICA SPC
EDITADO POR ERICSSON MEXICO 1985
- 9.-CENTRAL PROCESOR SUBSYSTEM
EDITADO POR ERICSSON STOCKHOLM 1988

PUBLICACIONES

- 1.-TELECOMMUNICATIONS
REVISTA MENSUAL USA OCTUBRE 1988
"ISDN CAN YOU AFFORD IT? "
- 2.-TELEPHONY
REVISTA MENSUAL USA JUNIO 22 1990
"SOFTWARE GLITCH CRIPPLES AT&T NETWORK"
- 3.-INTERNATIONAL BUSINESS WEEK
REVISTA SEMANAL USA MARCH 11 1991
"COMPUTER SOFTWARE : CAN THE US KEEP IT'S LEAD?"
- 4.-THE WALL STREET JOURNAL
DIARIO USA THURSDAY JUNE 27 1991
"REGIONAL PHONE SYSTEM DISRUPTED BY GLITCHES IS SOFTWARE"
- 5.-TELEPHONY
REVISTA MENSUAL USA JULY 01 1991
"SS7 ERRORS TORPEDO NETWORKS IN WASHINGTON, LOS ANGELES"

E203.03 LLAMADA EN ESPERA1. DEFINICION

Al estar un abonado en conversación, se le indica mediante un tono que otro abonado desea comunicarse con él. El abonado llamado tiene la posibilidad de retener la primera comunicación mientras atiende a la segunda, para más tarde regresar a su primera llamada.

2. EXPLICACION GENERAL (FUNCIONAMIENTO)

El estado del abonado en conversación es almacenado en una base de datos.

Cuando una llamada entrante esté intentando ocupar la línea en conversación, el sistema activa un temporizador para controlar el envío de tonos cortos cada 10 segundos.

Después del primer tono se tienen 30 segundos para atender la llamada o en caso contrario un programa SU se encargará de liberar la llamada. Ver figuras 1-2/E203.03.

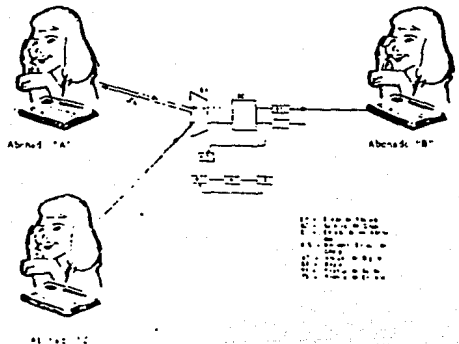


Fig. 1/E203.03 Diagrama Ilustrativo

**ESPECIFICACIONES TÉCNICAS DE
CENTRALES DIGITALES**
Fecha
88-04-07

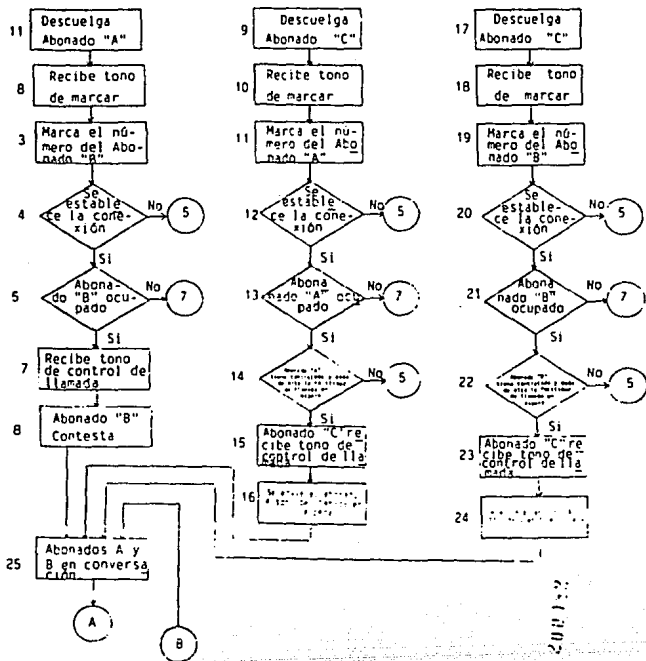
Edición
"A"


Fig. 2/E203 03 Especificación Operativa Función User

E203 03

**ESPECIFICACIONES TECNICAS DE
CENTRALES DIGITALES**

 Fecha
88-04-07

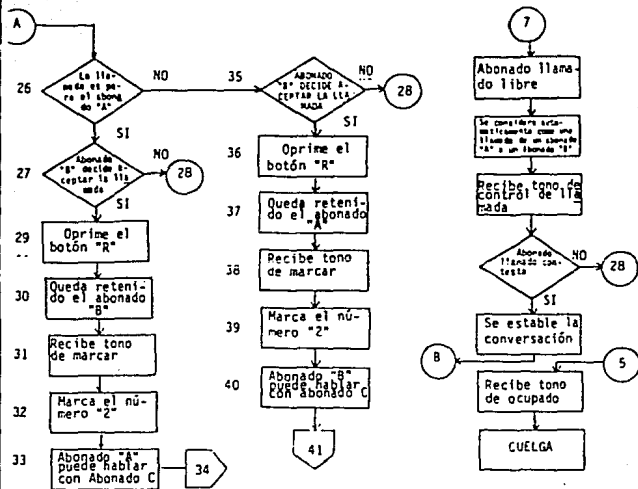
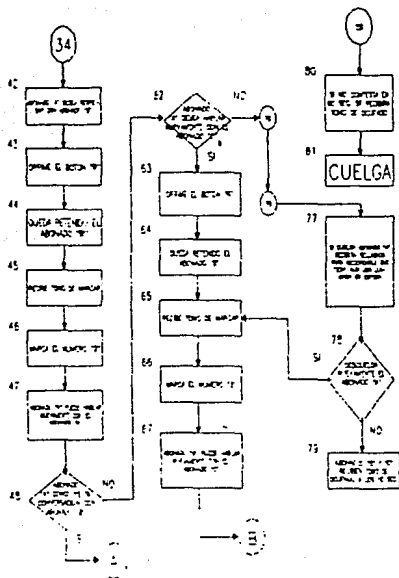
 Edición
"A"


Fig. 2/E203.03 Especificación Operativa Función Usar (Continuación)

E203.03

Fig. 2/E203 03 Especificación Operativa Función User
(Continuación).

ESPECIFICACIONES TÉCNICAS DE CENTRALES DIGITALES

Fecha
88-04-07

Edición
"A"

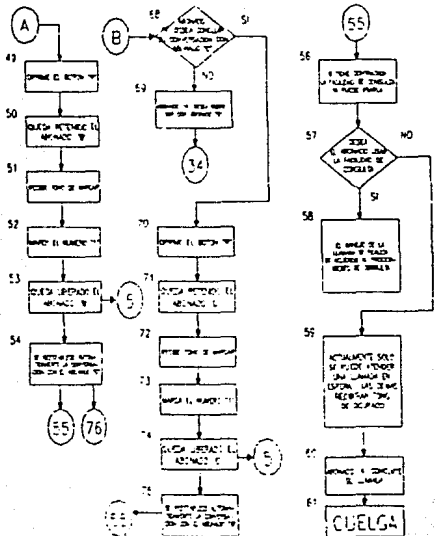


Fig. 2/E203:03 Especificación Operativa Función User (Continuación)

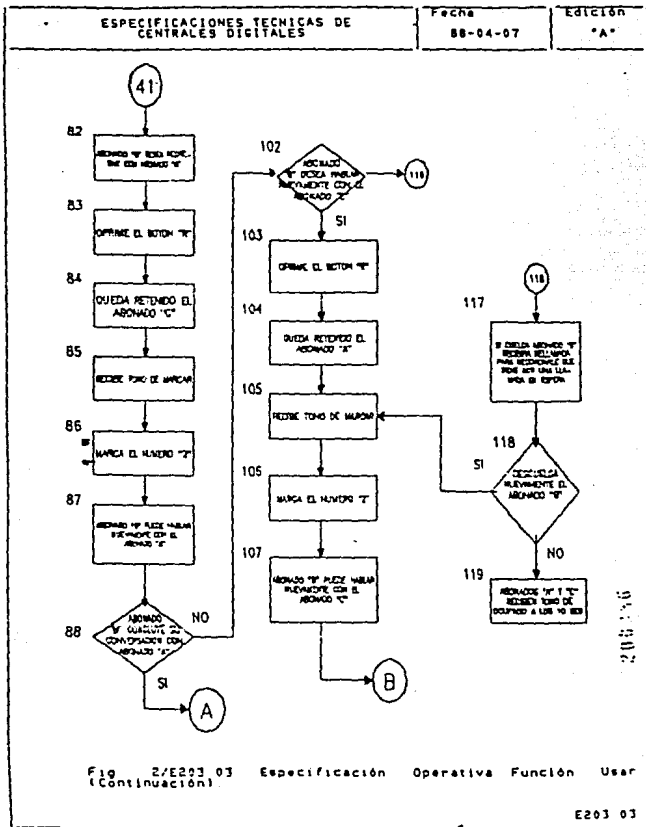
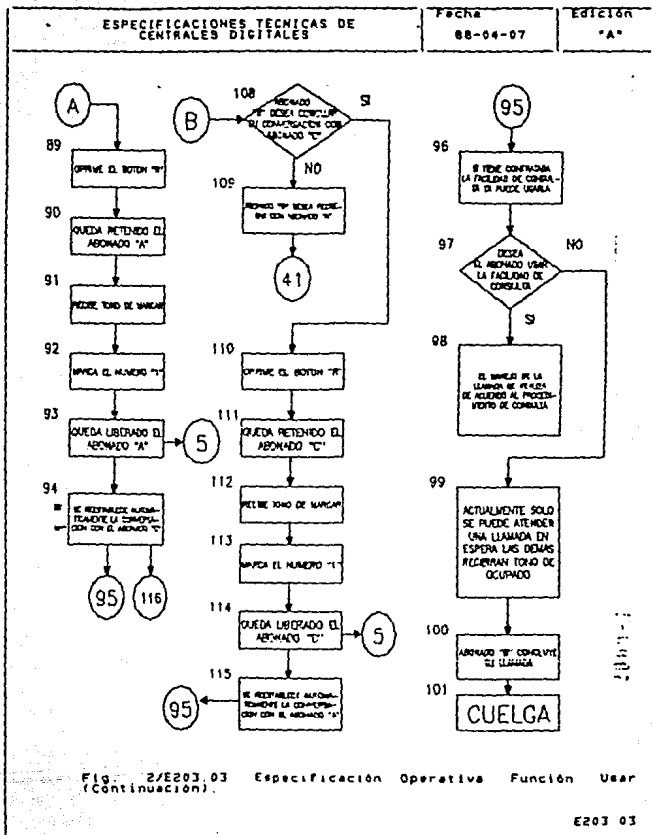
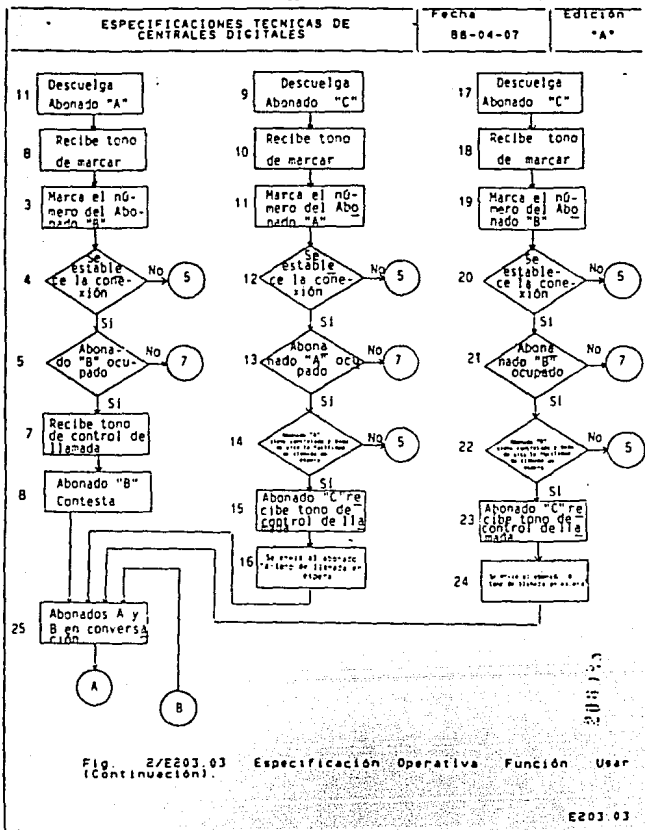


Fig. 2/E203 03 Especificación Operativa Función User (Continuación)





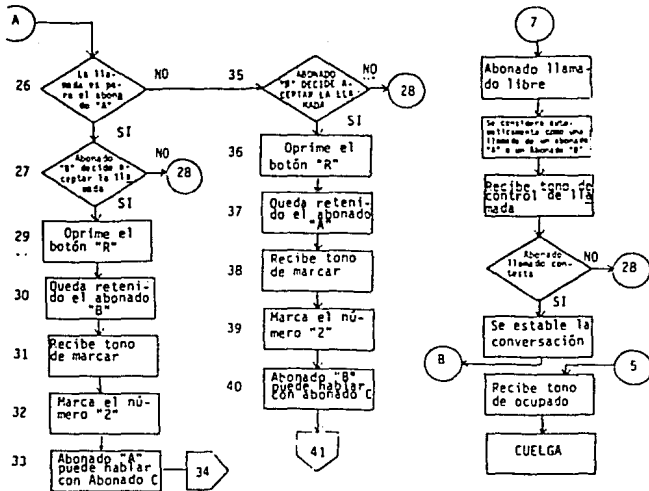
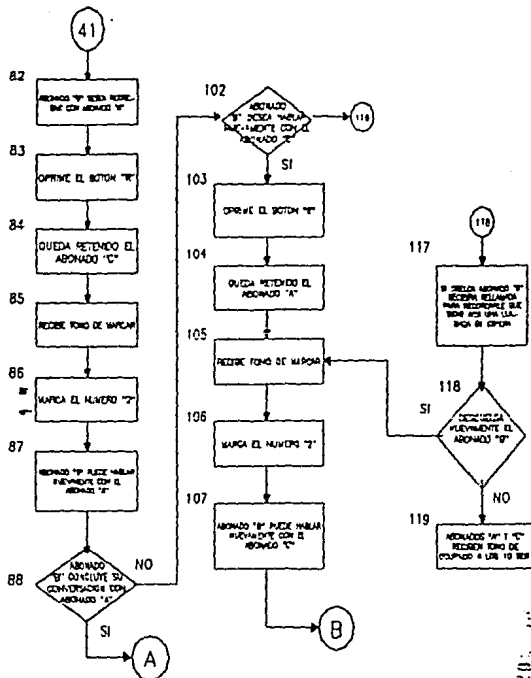


Fig. 2/E203.03 Especificación Operativa. Función User (Continuación).

Fig. 2/E203 03
(Continuación)

Especificación Operativa Función User

ESPECIFICACIONES TÉCNICAS DE CENTRALES DIGITALES

Fecha
88-04-07

Hoja: 2.31161
Edición
"A"

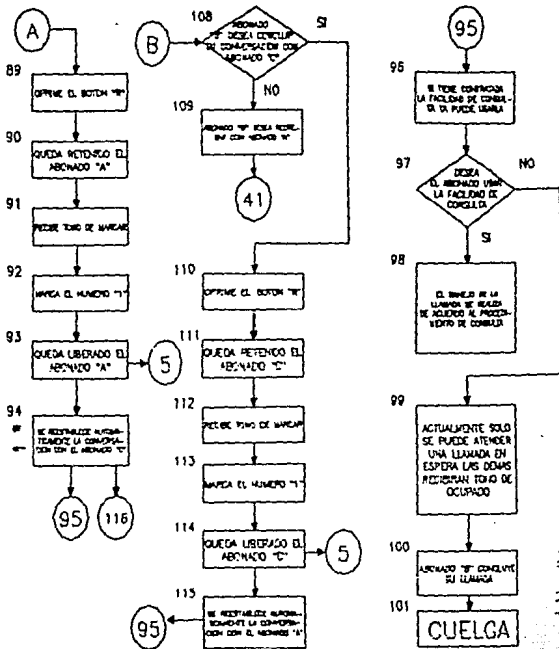


Fig. 2/E203.03 Especificación Operativa Función Usar (Continuación).

NSYMB CAWSTDSCITVALUE = 2;

PARAMETRO PRE-ESTABLECIDO

- 0 = SUBSCRITOR NO TIENE ACCESO A LOS PROCEDIMIENTOS (SCI=NONE)
- 1 = NO USADO
- 2 = SUBSCRITOR PUEDE CAMBIAR EL ESTADO DE ACTIVACION (SCI=ACT)
- 3 = NO USADO
- 4 = NO VALIDOO PARA CAW.
- 5-15 NO USADO
- VALOR NORMAL : 2
- RANGO DE VALOR : 0-15

PARAMETROS ALFANUMERICOS (DESIGNACION DEL PRODUCTO)

STRING CAWNAME = "LLAMADA EN ESPERA"

HOMBRE DE LA FACILIDAD
RANGO DE VALOR: 1C-3TC

STRING CAWABR = "CAW"

ABREVIACION DE LA FACILIDAD
RANGO DE VALOR: 7C-7C

STRING CWINAME="LLAMADA EN ESPERA ENTRANTE";

HOMBRE DE LA FACILIDAD
RANGO DE VALOR: 1C-3TC

STRING CUIABR = "CWI";

ABREVIACION DE LA FACILIDAD
RANGO DE VALOR: 7C-7C

END BLOCK;
END DOCUMENT;

NUMERO DE ERRORES = 0
NUMERO DE AVISOS = 26
COTICO DE ERROR = 0

DOCUMENT SUSCAMJSURVEY;
SUSCAMJSURVEY;
USE BLOCK SUSCAW;

```

SSSSSS LUJ LUJ SSSSSS CCCCCC AAAAAA WJ WJ
SSSSSS LUJ LUJ SSSSSS CC CCCCCC AA AAAAAA WJ WJ
SSSSSS LUJ LUJ SSSSSS C C CCCCCC AA AA WJ WJ
SSS LUJ LUJ SSS C C AAAAAA WJ WJ
SSS LUJ LUJ SSS C C AAAAAA WJ WJ
SSS LUJ LUJ SSS C C AAAAAA WJ WJ
SSSSSS LUJ LUJ SSSSSS CCCCCC AA AA WJ WJ
SSSSSS LUJ LUJ SSSSSS CCCCCC AA AA WJ WJ

```

```

LL LL SSSSSS TTTTTT AAAAAA
LL LL SSSSSS TTTTTT AA AA
LL LL SSSSSS TTTTTT AA AA
LL LL SSSSSS TTTTTT AA AA
LL LL SSSSSS TTTTTT AA AA
LL LL SSSSSS TTTTTT AA AA
LL LL SSSSSS TTTTTT AA AA

```

```

NNNNNN DDDDDDD EEEEEEE
DD DDD EEEEEEE
DD DDD EEEEEEE
DD DDD EEEEEEE
DD DDD EEEEEEE
DD DDD EEEEEEE
DD DDD EEEEEEE
DD DDD EEEEEEE

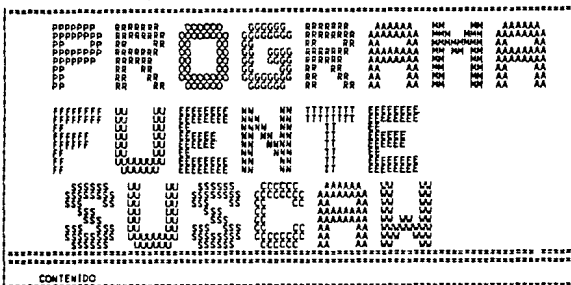
```

```

NN NN AAAAAA LL LL
NN NN AAAAAA LL LL
NN NN AA LL LL
NN NN AA LL LL
NN NN AAAAAA LL LL
NN NN AAAAAA LL LL
NN NN AA LL LL
NN NN AA LL LL
NN NN AA LL LL
NN NN AA LL LL

```


DOCUMENT SUSCAMPROGRAM;



CONTENIDO

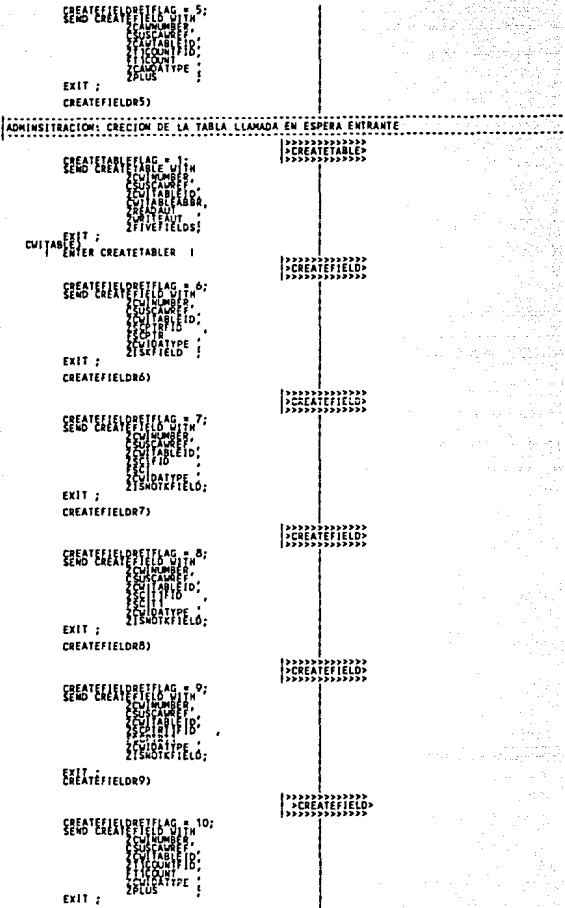
CONTENIDO:

1. PROLOGO
2. DECLARACION DE VARIABLES
 - 2.1. NOMBROS GLOBALES
 - 2.1.1. NOMBROS ALFANUMERICOS GLOBALES
 - 2.1.2. NOMBROS NUMERICOS GLOBALES
 - 2.1.1. NOMBROS NUMERICOS LOCALES
 - 2.2. ARCHIVOS COMUNES ALMACENADAS
 - 2.3. VARIABLES TEMPORALES
 - 2.4. VARIABLES ESTRUCTURADAS
3. PROGRAMA PRINCIPAL
 - 3.1. ENTRADA DE SEÑALES
 - 3.2. ARRANQUE Y REINICIO DEL SISTEMA
 - 3.2.1. RASTREO DE UNA LLAMADA AL REINICIO DEL SISTEMA
 - 3.3. ADMINISTRACION DE MEMORIA (ALTERACION 500)
 - 3.3.1. INCREMENTO
 - 3.4. ADMINISTRACION (MODIFICACION POR COMANDO)
 - 3.4.1. PROVISION DE LLAMADA EN ESPERA
 - 3.4.2. CANCELACION DE LLAMADA EN ESPERA
 - 3.4.3. CAMBIOS EN LLAMADA EN ESPERA
 - 3.4.4. MANEJO DE ERRORES EN LA ADMINISTRACION
 - 3.5. ADMINISTRACION DEL TRAFICO
 - 3.6. PROCEDIMIENTOS DEL SUBSCRIPCIÓN
 - 3.6.1. PROCEDIMIENTO POR EL PROGRAMA SUBSCRIPCIÓN
 - 3.6.2. PROCEDIMIENTO POR MEDIO DE UN SUBS. TERMINAL
 - 3.6.3. MANEJO DE ERRORES DURANTE LOS PROCEDIMIENTOS
 - 3.7. ESTABLECIMIENTO DE LA LLAMADA EN ESPERA
 - 3.7.1. ESTADO 1 DE LA LLAMADA
 - 3.7.2. ESTADO 2 DE LA LLAMADA
 - 3.7.3. ESTADO 3 DE LA LLAMADA
 - 3.7.4. ESTADO 4 DE LA LLAMADA
 - 3.7.5. ESTADO 5 DE LA LLAMADA
 - 3.7.6. ESTADO 6 DE LA LLAMADA
 - 3.7.7. ESTADO 7 DE LA LLAMADA
 - 3.7.8. ESTADO 8 DE LA LLAMADA
 - 3.7.9. ESTADO 9 DE LA LLAMADA
 - 3.7.10. ESTADO 10 DE LA LLAMADA
 - 3.7.11. ESTADO 11 DE LA LLAMADA
 - 3.8. RUTINAS COMUNES
 - 3.8.1. CONECTA DOS ABONADOS (DISRELEASELEG)
 - 3.8.2. LIBERA LÍNEA EN ESPERA (DISRELEASEINSER)
 - 3.8.3. DESCONECTA ABONADO (DISCONNECTION)
 - 3.8.4. MONITOREA ABONADOS (FREECORESENT)
 - 3.8.5. LIBERA CONEXIONES (FREERELEASELEG)
 - 3.9. SUBROUTINAS
 - 3.9.1. SELECCIONA INDIVIDUO LIBRE (IDLELISTFETCH)
 - 3.9.2. SELECCIONA INDIVIDUO LIBRE (IDLELISTINSERT)
 - 3.9.3. SEPARA INDIVIDUO LIBRE (SEPARATELIST)
 - 3.9.4. RECONSTRUYE LISTA DE INDIVIDUOS LIBRES (IDLELISTREBUILD)
 - 3.9.5. RECONSTRUYE SEGUNDA LISTA DE T. LIBRES (REBUILDTWOIDLELISTS)

- 3.9.6 MARCA INDIVIDUO (IDLEMARKTASKEINDIVIDUALS)
- 4. SECTOR DE DATOS
- 4.1 INICIALIZA ARCHIVOS
- 4.2 DATOS INICIALES
- 4.3 ALMACENAMIENTO DE VARIABLES

1.- PROLOGO	
<pre> PPPPPP RRRRRR PPPPPP RRRRRR PPPPPP RR RR PPPPPP RRRRRR PPPPPP RR RR PP RRRRRR PP RR RR PP RR RR </pre>	
<p>LLAMADA EN ESPERA (CV) es una facilidad de usuario que informa a un suscriptor que tiene una llamada esperando a ser contestada en la misma línea telefónica que él está usando en ese momento; el suscriptor puede entonces contestar la llamada que está en espera de dos maneras diferentes, la primera es marcar un código predeterminado de la manera que la llamada que está en espera es contestada y el suscriptor de la llamada original queda ahora en espera, la otra manera de contestar la llamada en espera es terminando la llamada original (colgando) y así el suscriptor es rellamado nuevamente por el suscriptor que estaba en espera, entonces cuando el suscriptor contesta esta nueva llamada, es conectado automáticamente con el suscriptor que estaba en espera...</p>	
2. DECLARACION DE VARIABLES	
DECLARE:	
2.1 SIMBOLOS GLOBALES	
2.1.1 SIMBOLOS GLOBALES NUMERICOS	
CATEGORIAS DE COMANDO	
GLOBAL NSYMB	<pre> BTXTSUSCAW (87777) BTXSUCAW (87777) BTXSUCVALVE (15) CAUSIDCVALDE (15) SUSCABLOCAT (255) </pre>
2.1.2 SIMBOLOS ALFANUMERICOS	
GLOBAL STRING	<pre> CAWABBR (57); "CAW" CWABBR (3); "Llamada en Espera" CVNAME (3); "Llamada en Espera Entrante" </pre>
2.2 SIMBOLOS LOCALES	
2.2.1 SIMBOLOS LOCALES NUMERICOS	
STRING	<pre> CAWTABLEABBR "CAWTB" DUMMT "SC1" DISCT1 "SC11" DISCT2 "SC12" DISCT3 "SC13" DISCT4 "SC14" DISCT5 "SC15" DISCT6 "SC16" DISCT7 "SC17" DISCT8 "SC18" DISCT9 "SC19" DISCT0 "SC10" DISCT10 "SC110" DISCT11 "SC111" DISCT12 "SC112" DISCT13 "SC113" DISCT14 "SC114" DISCT15 "SC115" DISCT16 "SC116" DISCT17 "SC117" DISCT18 "SC118" DISCT19 "SC119" DISCT20 "SC120" DISCT21 "SC121" DISCT22 "SC122" DISCT23 "SC123" DISCT24 "SC124" DISCT25 "SC125" DISCT26 "SC126" DISCT27 "SC127" DISCT28 "SC128" DISCT29 "SC129" DISCT30 "SC130" DISCT31 "SC131" DISCT32 "SC132" DISCT33 "SC133" DISCT34 "SC134" DISCT35 "SC135" DISCT36 "SC136" DISCT37 "SC137" DISCT38 "SC138" DISCT39 "SC139" DISCT40 "SC140" DISCT41 "SC141" DISCT42 "SC142" DISCT43 "SC143" DISCT44 "SC144" DISCT45 "SC145" DISCT46 "SC146" DISCT47 "SC147" DISCT48 "SC148" DISCT49 "SC149" DISCT50 "SC150" DISCT51 "SC151" DISCT52 "SC152" DISCT53 "SC153" DISCT54 "SC154" DISCT55 "SC155" DISCT56 "SC156" DISCT57 "SC157" DISCT58 "SC158" DISCT59 "SC159" DISCT60 "SC160" DISCT61 "SC161" DISCT62 "SC162" DISCT63 "SC163" DISCT64 "SC164" DISCT65 "SC165" DISCT66 "SC166" DISCT67 "SC167" DISCT68 "SC168" DISCT69 "SC169" DISCT70 "SC170" DISCT71 "SC171" DISCT72 "SC172" DISCT73 "SC173" DISCT74 "SC174" DISCT75 "SC175" DISCT76 "SC176" DISCT77 "SC177" DISCT78 "SC178" DISCT79 "SC179" DISCT80 "SC180" DISCT81 "SC181" DISCT82 "SC182" DISCT83 "SC183" DISCT84 "SC184" DISCT85 "SC185" DISCT86 "SC186" DISCT87 "SC187" DISCT88 "SC188" DISCT89 "SC189" DISCT90 "SC190" DISCT91 "SC191" DISCT92 "SC192" DISCT93 "SC193" DISCT94 "SC194" DISCT95 "SC195" DISCT96 "SC196" DISCT97 "SC197" DISCT98 "SC198" DISCT99 "SC199" DISCT100 "SC200" </pre>
NSYMB	<pre> CAW 11 CAW1 208 CAW2 50 CAW3 50 CAW4 50 CAW5 50 CAW6 50 CAW7 50 CAW8 50 CAW9 50 CAW10 50 CAW11 50 CAW12 50 CAW13 50 CAW14 50 CAW15 50 CAW16 50 CAW17 50 CAW18 50 CAW19 50 CAW20 50 CAW21 50 CAW22 50 CAW23 50 CAW24 50 CAW25 50 CAW26 50 CAW27 50 CAW28 50 CAW29 50 CAW30 50 CAW31 50 CAW32 50 CAW33 50 CAW34 50 CAW35 50 CAW36 50 CAW37 50 CAW38 50 CAW39 50 CAW40 50 CAW41 50 CAW42 50 CAW43 50 CAW44 50 CAW45 50 CAW46 50 CAW47 50 CAW48 50 CAW49 50 CAW50 50 CAW51 50 CAW52 50 CAW53 50 CAW54 50 CAW55 50 CAW56 50 CAW57 50 CAW58 50 CAW59 50 CAW60 50 CAW61 50 CAW62 50 CAW63 50 CAW64 50 CAW65 50 CAW66 50 CAW67 50 CAW68 50 CAW69 50 CAW70 50 CAW71 50 CAW72 50 CAW73 50 CAW74 50 CAW75 50 CAW76 50 CAW77 50 CAW78 50 CAW79 50 CAW80 50 CAW81 50 CAW82 50 CAW83 50 CAW84 50 CAW85 50 CAW86 50 CAW87 50 CAW88 50 CAW89 50 CAW90 50 CAW91 50 CAW92 50 CAW93 50 CAW94 50 CAW95 50 CAW96 50 CAW97 50 CAW98 50 CAW99 50 CAW100 50 </pre>

1	...	0
2	...	0
3	...	0
4	...	0
5	...	0
6	...	0
7	...	0
8	...	0
9	...	0
10	...	0
11	...	0
12	...	0
13	...	0
14	...	0
15	...	0
16	...	0
17	...	0
18	...	0
19	...	0
20	...	0
21	...	0
22	...	0
23	...	0
24	...	0
25	...	0
26	...	0
27	...	0
28	...	0
29	...	0
30	...	0
31	...	0
32	...	0
33	...	0
34	...	0
35	...	0
36	...	0
37	...	0
38	...	0
39	...	0
40	...	0
41	...	0
42	...	0
43	...	0
44	...	0
45	...	0
46	...	0
47	...	0
48	...	0
49	...	0
50	...	0
51	...	0
52	...	0
53	...	0
54	...	0
55	...	0
56	...	0
57	...	0
58	...	0
59	...	0
60	...	0
61	...	0
62	...	0
63	...	0
64	...	0
65	...	0
66	...	0
67	...	0
68	...	0
69	...	0
70	...	0
71	...	0
72	...	0
73	...	0
74	...	0
75	...	0
76	...	0
77	...	0
78	...	0
79	...	0
80	...	0
81	...	0
82	...	0
83	...	0
84	...	0
85	...	0
86	...	0
87	...	0
88	...	0
89	...	0
90	...	0
91	...	0
92	...	0
93	...	0
94	...	0
95	...	0
96	...	0
97	...	0
98	...	0
99	...	0
100	...	0
101	...	0
102	...	0
103	...	0
104	...	0
105	...	0
106	...	0
107	...	0
108	...	0
109	...	0
110	...	0
111	...	0
112	...	0
113	...	0
114	...	0
115	...	0
116	...	0
117	...	0
118	...	0
119	...	0
120	...	0
121	...	0
122	...	0
123	...	0
124	...	0
125	...	0
126	...	0
127	...	0
128	...	0
129	...	0
130	...	0
131	...	0
132	...	0
133	...	0
134	...	0
135	...	0
136	...	0
137	...	0




```

TSTOPSCANNING = TSTARTSCANNING + (ZMAYSCAN-1);
FI;
ON TASKCAMP FROM TSTARTSCANNING UPTO TSTOPSCANNING DO
DO SEPARATELELE181;
NO;
IF TSTOPSCANNING /= TNEWINDMUM - 1 THEN I
SCANNING = TSTOPSCANNING - 1;
SEND COM MUEC REFERENCE CSUSCAREF WITH I
STATE INCREASE20,
SCANNING,
TNEWINDMUM;
EXIT;
EL CPREPNUM = TNEWINDMUM;
CINDMUM = TNEWINDMUM;
IF TINDLE = 0;
SEND SELFEND;
EXIT;
FI;

```

```

-----
CHECA SI FUE ALTERADO ANTERIORMENTE
SAINCREASE30)
TSCANNING = CPREPNUM;
SAINCREASE30)
TSTOPSCANNING = TSCANNING;
TSTOPSCANNING = CINDMUM;
IF TSTOPSCANNING - TSTARTSCANNING > (ZMAYSCAN-1) THEN
FI;
ON TASKCAMP FROM TSTARTSCANNING UPTO TSTOPSCANNING DO
IF STATE = IDLE THEN
DO IDLELISTAREBUILD;
NO;
IF TSTOPSCANNING /= CINDMUM - 1 THEN
SCANNING = TSTOPSCANNING - 1;
SEND COM MUEC REFERENCE CSUSCAREF WITH
STATE INCREASE20,
SCANNING,
TNEWINDMUM;
EXIT;
EL SCANNING = CINDMUM;
FIRST = 0;
GOTO SAINCREASE20;
FI;

```

3.4 ADMINISTRACION MODIFICACION POR COMANDO DE LA FACILIDAD LLAMADA EN ESPERA

AAAAAA	DDDDDD	NN	NN		NN	NN		SSSSSS
AAAAAAA	DDDDDD	NNN	NNN		NNN	NNN		SSSSSS
AA	AA	NNN	NNN		NNN	NNN		SSSSSS
AAAAAAA	DD	NN	NN		NN	NNN		SSSSSS
AAAAAAA	DD	NNN	NN		NN	NNNN		SSSSSS
AA	AA	NNN	NN		NN	NNNN		SSSSSS
AA	DDDDDD	NN	NN		NN	NN		SSSSSS
AA	DDDDDD	NN	NN		NN	NN		SSSSSS

	RRRRRR	AAAAAA	CCCCC		OOOOO	NN	NN
	RRRRRR	AAAAAAA	CCCCC		OOOOO	NNN	NN
	RRRRRR	AAAAAAA	CC		OOOOO	NNN	NN
	RRRRR	AAAAAAA	CC		OOOOO	NN	NNN
	RR	RR	AA	AA	CCCCC	NN	NN
	RR	RR	AA	AA	CCCCC	NN	NN

```

ADMINIA)
CASE FUNCTION IS
WHEN 0 DO
GOTO PROVIDE10;
WHEN 1 DO
GOTO CHANGE10;
WHEN 2 DO
GOTO REMOVE10;
OTHERWISE DO
GOTO EXIT;
END;

```

3.4.1 PROVISION DE LLAMADA EN ESPERA

pppppp	RRRRRR	OOOOO	VV	EEEEEE	EEEEEE	RRRRRR
ppppppp	RRRRRRR	OOOOO	VV	EEEEEE	EEEEEE	RRRRRRR
pp	RR	RR	VV	VV	EEEEEE	RR
pppppp	RRRRRRR	OOOOO	VV	VV	EEEEEE	RRRRRRR
pppppp	RRRRR	OOOOO	VV	VV	EEEEEE	RRRRR
pp	RR	RR	VV	VV	EEEEEE	RR
pp	RR	RR	VV	VV	EEEEEE	RR
pp	RR	RR	VV	VV	EEEEEE	RR

```

PROVIDE10) C = ADMINPROVIDE;
UPDATEETFLAG = 0;
|||||
>RECEIVEINF>
>|||||

```



```

TABLEID
ISCTYID ,
ZPLUS ,
ZPLUS ,
ZPLUS ,
ZPLUS ,
ZONEFIELD ,
ZMMY
ZICOUNTFID ;

EXIT;
REMOVE0)
IF RESULTCODE = ZINVALIDKF THEN
  FAULT = ZSTDFC10 ;
  GOTO ADMINERR ;
IF RESULTCODE /= ZSUCCESS THEN
  FAULT = ZSTDFC11 ;
  GOTO ADMINERR ;
FI;
TICOUNTERT1 = TRETREIVAL1;
                                     >>>>>>>>>>>>
                                     >RETRIEVEINF<
                                     >>>>>>>>>>>>

RETRIEVEREFLAG = 3 ;
SEND RETRIEVEINF REFERENCE FSHREF WITH
  FSHPR
  TASCAM0'
  TASCAM0'
  CSUSCAM0F ,
  TRANSACTIONID ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZONEFIELD ,
  ZMMY
  ZCAMPROPERTY ;

EXIT;
REMOVE1)
IF RESULTCODE = ZINVALIDKF THEN
  FAULT = ZSTDFC10 ;
  GOTO ADMINERR ;
IF RESULTCODE /= ZSUCCESS THEN
  FAULT = ZSTDFC11 ;
  GOTO ADMINERR ;
IF PROPERACC = TRETREIVAL1
  IF PROPERACC /= 0 GOTO REMOVE0 ;
                                     >>>>>>>>>>>>
                                     >RETRIEVEINF<
                                     >>>>>>>>>>>>

RETRIEVEREFLAG = 4 ;
SEND RETRIEVEINF REFERENCE FSHREF WITH
  FSHPR
  TASCAM0'
  CSUSCAM0F ,
  TRANSACTIONID ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZONEFIELD ,
  ZMMY
  ZCAMPROPERTY ;

EXIT;
REMOVE2)
IF RESULTCODE = ZINVALIDKF THEN
  FAULT = ZSTDFC10 ;
  GOTO ADMINERR ;
IF RESULTCODE /= ZSUCCESS THEN
  FAULT = ZSTDFC11 ;
  GOTO ADMINERR ;
IF PROPERACC = TRETREIVAL1 ;
REMOVE0)
                                     >>>>>>>>>>>>
                                     >UPDATEINF >
                                     >>>>>>>>>>>>

UPDATEREFLAG = 0 ;
SEND UPDATEINF REFERENCE FSHREF WITH
  FSHPR
  TASCAM0'
  TASCAM0'
  CSUSCAM0F ,
  TRANSACTIONID ,
  CSUSCAM0F ,
  TRANSACTIONID ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZPLUS ,
  ZONEFIELD ,
  ZMMY
  ZCAMPROPERTY ;

```

```

                                ZPLUS
                                ONEFIELD
                                CUMM1
                                PROPERTY
                                ZDEACT
EXIT;

REMOVE0)
UPDATERETFLAG = 1 GOTO REMOVE0;
UPDATERETFLAG = 2 GOTO REMOVE0;
UPDATERETFLAG = 3 GOTO REMOVE120;

IF RESULTCODE = ZOUTOFRANGE THEN
    FAULT = ZSTDF10;
    GOTO ADMINERR;
[[ RESULTCODE /# ZSUCCESS THEN
    FAULT = ZSTDF11;
    GOTO ADMINERR
]]
[[ COUNTER1 > 0 GOTO REMOVE0;
]]
                                >>>>>>>>>>
                                >DELETEINF >
                                >>>>>>>>>>

DELETERETFLAG = 0;
SEND DELETEINF REFERENCE FSHREF WITH
    SMPTR
    TASCAMP
    TASCAMREF
    TASCAMREF
    TASCAMREF
    TABLEID
    SCPTRT
    SCPTRT
    PLUS
    PLUS
    PLUS
    ONEFIELD,
    CUMM1
    SCPTRT
    SCPTRT
    SCPTRT
    SCPTRT
    SCPTRT
    ZNULL;

EXIT;
REMOVE0)
                                >>>>>>>>>>
                                >UPDATEINF >
                                >>>>>>>>>>

UPDATERETFLAG = 1;
SEND UPDATEINF REFERENCE FSHREF WITH
    SMPTR
    TASCAMP
    TASCAMREF
    TASCAMREF
    TASCAMREF
    TABLEID
    SCPTRT
    SCPTRT
    SCPTRT
    SCPTRT
    SCPTRT
    PLUS
    PLUS
    PLUS
    ONEFIELD,
    CUMM1
    SCPTRT
    SCPTRT
    SCPTRT
    SCPTRT
    SCPTRT
    ZNULL;

EXIT;
REMOVE0)
DELETERETFLAG = 1 GOTO REMOVE120;
IF RESULTCODE = ZOUTOFRANGE THEN
    FAULT = ZSTDF10;
    GOTO ADMINERR;
[[ RESULTCODE /# ZSUCCESS THEN
    FAULT = ZSTDF11;
    GOTO ADMINERR;
]]
[[ SCPTRT1 = ZNULL GOTO REMOVE120;
]]
[[ COUNTER1 = TCOUNTER1 - 1;
]]
[[ TCOUNTER1 = 0 GOTO REMOVE100;
]]
                                >>>>>>>>>>
                                >UPDATEINF >
                                >>>>>>>>>>

UPDATERETFLAG = 2;
SEND UPDATEINF REFERENCE FSHREF WITH
    SMPTR
    TASCAMP
    TASCAMREF
    TASCAMREF
    TASCAMREF
    TABLEID
    SCPTRT
    SCPTRT
    SCPTRT
    SCPTRT
    SCPTRT
    PLUS
    PLUS
    PLUS
    ONEFIELD
    CUMM1
    TCOUNTER1
    TCOUNTER1;

```



```

-----
OBTIENE INFORMACION ACERCA DEL SUBSCRIPTOR TERMINAL
-----
[MASK = VALIDMASC(180000);
IF MASK = 0000 THEN
GOTO CHANGE70 ;
FI;

RETRIEVEFLAG = 2 ;
SEND RETRIEVEINF REFERENCE FSHREF WITH
    FSHRT
    TASKAMP
    SUSCALREF
    TRANACTIO
    PLUS
    SUSCALREF
    TABLEID
    COUNTRY
    DISCPTRI1
    PLUS
    PLUS
    PLUS
    ONEFIELD
    COUNTRY
    TICOUNTFID ;

EXIT;

CHANGE60)
IF RESULTCODE = 2SUCCESS THEN
RECORD = 2SUCCESS ;
ELSE
RESULTCODE = 2INVALIDKF THEN
RECORD = 2NOTFOLND;
ELSE
FAULT = 2STDFCI1;
GOTO ADMINERR ;
FI;
TICOUNTERT1 = RETRIEVALT ;
-----
OBTIENE LOS DATOS ALMACENADOS EN LAS BASES DE DATOS PARA ESE SUBSCRIPTOR
CHANGE70)
IF OLDSCTRI1=2NULL GOTO CHANGE100;

RETRIEVEFLAG = 3 ;
SEND RETRIEVEINF REFERENCE FSHREF WITH
    FSHRT
    TASKAMP
    SUSCALREF
    TRANACTIO
    PLUS
    SUSCALREF
    PROPERTYB
    PROPERTYC
    DISCPTRI1
    PLUS
    PLUS
    PLUS
    ONEFIELD
    COUNTRY
    PROPERTY ;

EXIT;

CHANGE80)
IF RESULTCODE = 2INVALIDKF THEN
FAULT = 2STDFCI1;
GOTO ADMINERR ;
IF RESULTCODE /= 2SUCCESS THEN
FAULT = 2STDFCI1;
GOTO ADMINERR ;
FI;

OLDTIACC = RETRIEVALT ;
-----
RETRIEVEFLAG = 4 ;
SEND RETRIEVEINF REFERENCE FSHREF WITH
    FSHRT
    TASKAMP
    SUSCALREF
    TRANACTIO
    PLUS
    SUSCALREF
    TABLEID
    OLDSCTRI1
    PLUS
    PLUS
    PLUS
    ONEFIELD
    COUNTRY
    TICOUNTFID;

EXIT;
-----

```



```

CHANGE90)
  ( RESULTCODE = ZINVALIDKF THEN
  FAIL = #000 THEN
  GOTO ADMINERR ;
  ( RESULTCODE /# SUCCESS THEN
  FAIL = #000 THEN
  GOTO ADMINERR ;
  FI;
  COLDTCOUNT11 = TRETREIVAL1 ;
  
```

```

CHANGE100)
  IMASK = VALIMASK(') #000;
  IF IMASK = #000 THEN
  SIFSCTRT1 = OLDSCTRT1 ;
  GOTO CHANGE170 ;
  ( SIFSCTRT1 = OLDSCTRT1
  GOTO CHANGE170 ;
  IF SIFSCTRT1 = SIFSCTRT1 THEN
  ( OLDSCTRT1 /# ZNULL THEN
  SIFSCTRT1 = ZNONE
  UPDATECAUTABLE = ZTRUE;
  GOTO CHANGE170 ;
  ELSE
  GOTO CHANGE170 ;
  FI;
  ELSE
  UPDATECAUTABLE = ZTRUE;
  FI;
  
```

```

  IMASK = VALIMASK(') #000;
  IF IMASK = #000 THEN
  SIFSCT1 = CAWSTOSCI1VALUE;
  FI;
  IF RECORD = ZNOTFOUND GOTO CHANGE120;
  TCOUNTERT1 = TICOUNTERT1 + 1 ;
  
```

```

>>>>>>>>>>>>
>>>>>>>>>>>>
>>>>>>>>>>>>
  
```

```

  UPDATERETFLAG = 0;
  SEND UPDATEINF REFERENCE FSHREF WITH
  ( SHPTR
  ( TASKCALP
  ( CSUSCAUREF
  ( TRANSACTIO
  ( CSUSCAUREF
  ( TABLEID
  ( CAWSTOSCI1
  ( SIFSCTRT1
  ( PLUS
  ( PLUS
  ( PLUS
  ( NONEFIELD
  ( NONE
  ( TICOUNTERT1
  ( TICOUNTERT1 ;
  
```

EXIT;

CHANGE120)

```

>>>>>>>>>>>>
>>>>>>>>>>>>
>>>>>>>>>>>>
  
```

```

  SEND INSERTINF REFERENCE FSHREF WITH
  ( SHPTR
  ( TASKCALP
  ( CSUSCAUREF
  ( TRANSACTIO
  ( CSUSCAUREF
  ( TABLEID
  ( CAWSTOSCI1
  ( SIFSCTRT1
  ( PLUS
  ( PLUS
  ( PLUS
  ( NONEFIELD,
  ( NONE
  ( NONEID ;
  ( NONEID ;
  ( NONEID ;
  ( SCTR1FID
  ( SCTR1FID
  ( SCTR1FID
  ( SCTR1FID
  ( SCTR1FID ;
  
```

CHANGE130)

```

  IF UPDATERETFLAG = 1 GOTO CHANGE160;
  IF UPDATERETFLAG = 2 GOTO CHANGE200;
  IF UPDATERETFLAG = 3 GOTO ADMINEND10;
  IF RESULTCODE = ZOUTOFRANGE THEN
  
```

```

FAULT = ZSDFC10;
GOTO ADMINERR ;
FI;
IF RESULTCODE /= ZSUCCESS THEN
FAULT = ZSDFC11;
GOTO ADMINERR ;
FI;
IF OLDSCTRT1 = #FFFF GOTO CHANGE170;

```

```

CHANGE140)
COLDTCOUNT11 = COLDTCOUNT11 - 1;
IF COLDTCOUNT11 >= 0 GOTO CHANGE150;
IF OLDTIACC /= 0 GOTO CHANGE150;

```

```

SEND DELETEINF REFERENCE FSHREF WITH
FSHPTF
TASKCAID*
ZSCSCAUREF*
TRANSACTION*
COLDTCOUNT11*
TABLEID*
CAWVTC*
ZCAWVTC*
OLDSCTRT1;

```

```

EXIT;
CHANGE150)

```

```

UPDATETEIAC = 1;
SEND UPDATEINF REFERENCE FSHREF WITH
FSHPTF
TASKCAID*
ZSCSCAUREF*
TRANSACTION*
COLDTCOUNT11*
TABLEID*
CAWVTC*
ZCAWVTC*
ZSUSCSTR*
ZPLUS*
ZPLUS*
ZPLUS*
ZONEFIELD*
ZMM*
ZCOUNTID*
ZCOUNTER11 ;

```

```

EXIT;
CHANGE160)
IF RESULTCODE = ZOUTDFRANGE THEN
FAULT = ZSDFC10;
GOTO ADMINERR ;
FI;
IF RESULTCODE /= ZSUCCESS THEN
FAULT = ZSDFC11;
GOTO ADMINERR ;
FI;

```

```

CHANGE170)

```

```

IMASK = VALIDMASK(*)#0010;
IF IMASK = #0000 THEN;
SUSC11 = 0;
GOTO CHANGE180 ;
FI;
SUSC11 /= OLDSCT11 THEN
UPDATECAWVTABLE = ZTRUE;
CASE SUSC11 IS
WHEN 0 DO
WHEN SUSC11 = ZMOMESCI;
WHEN SUSC11 = CAWSTDSC11VALUE;
WHEN SUSC11 = ZACTSCI ;
OTHERWISE DO
FAULT = ZSDFC21;
GOTO ADMINERR ;
ESAC;
FI;

```

```

CHANGE180)

```

```

IMASK = VALIDMASK(*)#0008;
IF IMASK = #0000 THEN;
SUSC1 = 0;
GOTO CHANGE190 ;
FI;
SUSC1 /= OLDSCT1 THEN
UPDATECAWVTABLE = ZTRUE;

```

```

CASE SUFSCI IS
WHEN 0 DO
  SUFSCI = ZWOMESCI ;
WHEN 1 DO
  SUFSCI = CAWSTDSCIVALUE;
WHEN 2 DO
  SUFSCI = ZACTSICI ;
OTHERWISE DO
  SUFSCI = ZSTDFC14;
GOTO ADMINERR;
ESAC;
FI;
-----
SOLICITUD DE ACTIVAR
CHANGE100)
  TTASK = VALIDMASC(*)#0200;
  IF TASK = #0000 THEN
    GOTO CHANGE200;
  VALUE = ZACT;
  GOTO CHANGE210;
-----
SOLICITUD DE DESACTIVAR
CHANGE200)
  TTASK = VALIDMASC(*)#0400;
  IF TASK = #0000 THEN GOTO CHANGE220 ;
  VALUE = ZDEACT;
  GOTO CHANGE210;
-----
UPDATESETFLAG = 2;
SEND UPDATEINF REFERENCE FSHREF WITH
  TASKAMP;
  CSUSCAMP;
  TRANSACTION;
  SUFSCREF;
  SUTIDGER;
  SUTIDGERVIEW;
  SUTIDGER;
  PLUS;
  PLUS;
  PLUS;
  THREEFIELDS;
  NUMBER;
  PROPERTY;
  VALUE;
EXIT;
CHANGE220)
  IF RESULTCODE = ZOUTOFRANGE THEN
    FAIL = ZSTDFC19;
    GOTO ADMINERR;
  FI;
  IF RESULTCODE /= ZSUCCESS THEN
    FAIL = ZSTDFC11;
    GOTO ADMINERR;
  FI;
  IF UPDATECAVTABLE /= ZTRUE
    GOTO ADMINEND10;
-----
UPDATESETFLAG = 3;
SEND UPDATEINF REFERENCE FSHREF WITH
  TASKAMP;
  CSUSCAMP;
  TRANSACTION;
  CSUSCAMP;
  TABLEID;
  TABLEID;
  SUTIDGER;
  PLUS;
  PLUS;
  PLUS;
  THREEFIELDS;
  NUMBER;
  SUTIDGER;
  SUTIDGER;
  SUTIDGER;
  SUTIDGER;
  SUTIDGER;
  SUTIDGER;
  SUTIDGER;
EXIT;
-----
RUTINA COMUN PARA LA PARTE DE ADMINISTRACION EN CASO DE SER EXITOSO EL
PROCEDIMIENTO REALIZADO
ADMINEND10)
  IF RESULTCODE = ZOUTOFRANGE THEN
    FAIL = ZSTDFC19;
    GOTO ADMINERR;
  FI;
  IF RESULTCODE /= ZSUCCESS THEN
    FAIL = ZSTDFC11;
    GOTO ADMINERR;
  FI;
-----

```

```

        >>>>>>>>>>
        >>>>>>>>>>
        SEND PREPARE REFERENCE FSHREF WITH
          FSHPR ;
          TASKCAUP ;
          CSUSCAUREF ;
          TRANSACTIO ;
        EXIT;
ADMINEND20)
IF RESULTCODE = ZRECORDLCKD THEN
  FAULT = ZSTDFCT1 ;
  GOTO ADMINERR ;
ELSE
  IF RESULTCODE /= ZSUCCESS THEN
    FAULT = ZSTDFCT1 ;
    GOTO ADMINERR ;
  FI;
        >>>>>>>>>>
        >>>>>>>>>>
        SEND COMMIT REFERENCE FSHREF WITH
          FSHPR ;
          TASKCAUP ;
          CSUSCAUREF ;
          TRANSACTIO ;
        EXIT;
ADMINEND30)
ACTUALIZA CONTADORES
IF FEATNUM = ZCAWNUMBER THEN
  SUSCOUNTERP = 0 ;
  COUNTERSUBSACC = COUNTERSUBSACC + 1 ;
  ELSE
  SUSCOUNTERP = 1 ;
  COUNTERSUBSACC = COUNTERSUBSACC + 1 ;
  FI;
GOTO ADMINEND60;
ADMINEND40)
IF FEATNUM = ZCAWNUMBER THEN
  SUSCOUNTERP = 0 ;
  COUNTERSUBSACC = COUNTERSUBSACC - 1 ;
  ELSE
  SUSCOUNTERP = 1 ;
  COUNTERSUBSACC = COUNTERSUBSACC - 1 ;
  FI;
IF VALUE = ZDEACT THEN
  IF FEATNUM = ZCAWNUMBER THEN
    SUSCOUNTERP = COUNTERSUBSACC - 1 ;
    COUNTERSUBSACC = COUNTERSUBSACC - 1 ;
  ELSE
    SUSCOUNTERP = 1 ;
    COUNTERSUBSACC = COUNTERSUBSACC - 1 ;
  FI;
GOTO ADMINEND60;
ADMINEND50)
IF VALUE = ZACT THEN
  IF FEATNUM = ZCAWNUMBER THEN
    SUSCOUNTERP = COUNTERSUBSACC + 1 ;
    COUNTERSUBSACC = COUNTERSUBSACC + 1 ;
  ELSE
    SUSCOUNTERP = 1 ;
    COUNTERSUBSACC = COUNTERSUBSACC + 1 ;
  FI;
IF VALUE = ZDEACT THEN
  IF FEATNUM = ZCAWNUMBER THEN
    SUSCOUNTERP = 0 ;
    COUNTERSUBSACC = COUNTERSUBSACC - 1 ;
  ELSE
    SUSCOUNTERP = 1 ;
    COUNTERSUBSACC = COUNTERSUBSACC - 1 ;
  FI;
ADMINEND60)
ENVIA MENSAJE DE QUE EL COMANDO FUE EJECUTADO
STATE = ADMINEND;
        >>>>>>>>>>>>>>
        >>>>>>>>>>>>>>
        SEND SENDINF REFERENCE FSHREF WITH
          FSHPR ;
          TASKCAUP ;
          CSUSCAUREF ;
          TRICOLEG ;
          ZRESINFO ;
          ZELEMENTO ;
        EXIT;
ADMINEND70)
        >>>>>>>>>>>>>>
        >>>>>>>>>>>>>>
        SEND FREE REFERENCE FSHREF WITH
          FSHPR ;
          TASKCAUP ;
          CSUSCAUREF ;

```



```

SEND STARTTRANSACT REFERENCE FSHREF WITH
  FSHPTR
  TASKCAWD
  CSUSCAWREF;
EXIT;
SUBSPROC0)
IF RESULTCODE = #ZSUCCESS THEN
  RESULTCODE = #ZUNEXPECTED;
GOTO SUBSPERR20;
FI;
TRANSACTION = ITRANSACTION ;
GOTO SUBSPROC50 ;

```

=====

3.6.2 PROCEDIMIENTO DE USUARIO REALIZADO POR EL SUBSCRIPTOR TERMINAL

=====

```

T1SUBSPROC10)
STATE = #T1SUBSPROC;

SEND SENDRECEIVEINFR REFERENCE FSHREF WITH !
  FSHPTR
  TASKCAWD
  CSUSCAWREF
  ALFCID
  DIALOGINF
  MESSONL
  #PLUS
  #MVSINFO
  #MRESSONL
  #ZELEMENT11;
EXIT;
>>>>>>>>>>>>>>
>SENDRECEIVEINFR>
>>>>>>>>>>>>>>
ENTER SENDRECEIVEINFR WITH
  FSHPTR
  FSHREF
  RESULTCODE;

IF RESULTCODE = #ZSUCCESS THEN
  RESULTCODE = #ZUNEXPECTED;
GOTO SUBSPERR20;
FI;

T1SUBSPROC20)
ALFGONMASK = #ZCLEARED;
BLEFGONMASK = #ZCLEARED;
CLEFGONMASK = #ZCLEARED;
RESULT = #ZSUCCESS;
ALFGONMASK ONHOOK = #ZSET;
STATE = #RECALLTALONE;
GOTO MONITOREVENT;

```

=====

ANALISIS DE LOS DIGITOS QUE MARCO EL TERMINAL

=====

```

T1RDTONE10)
CASE TEVENT IS
WHEN ZINFCOANALYT DO
STATE = #T1SUBSPROC;
GOTO T1SUBSPROC30;
WHEN ZONHOOK DO
RESULT = #ZSUCCESS;
ALFCITFREE = #TRUE;
GOTO DIRECTION;
WHEN ZLEGITSC DO
RESULT = #ZSUCCESS;
GOTO T1WISH;
OTHERWISE DO
GOTO T1SUBSPROC20;
ESAC;

```

=====

RESULTADOS DEL ANALISIS DE DIGITOS

=====

```

T1SUBSPROC30)
>>>>>>>>>>>>>>
>RECEIVEINFR>
>>>>>>>>>>>>>>
SEND RECEIVEINFR REFERENCE FSHREF WITH
  FSHPTR
  TASKCAWD
  CSUSCAWREF;
  ALFCID
  DIALOGINF
  #MVSITIVE;
EXIT;
T1SUBSPROC40)
TRASC = #T1SUBSPROC2 (*) #D008;

```

```

IF TRASC = #0008 THEN
  ESMES = #01INVALID ;
  GOTO SUBSPERR20 ;
T1SCPTR = TRCEIVEINF5 ;
-----
INICIA DIALOGO CON BASE DE DATOS
-----
                                >>>>>>>>>>
                                >STARTRANSACT>
                                >>>>>>>>>>
SEND STARTRANSACT REFERENCE FSHREF WITH
  FSHREF
  TASCALUP
  CSUSCALUPREF;
EXIT;
T1SUBSPROC50)
IF RESULTCODE /= #ZSUCCESS THEN
  ESMES = #01INVALID ;
  GOTO SUBSPERR20 ;
FI;
TRANSACTID = TTRANSACTID ;
IF SCPTR = T1SCPTR
  GOTO SUBSPROC50 ;
-----
OBTIENE EL APUNTAJADOR DEL ARCHIVO DEL SUBSCRIPTOR TERMINAL1
-----
                                >>>>>>>>>>
                                >RETRIEVEINF>
                                >>>>>>>>>>
RETRIEVEREF1AG = 0;
SEND RETRIEVEINF REFERENCE FSHREF WITH
  FSHREF
  TASCALUP
  CSUSCALUPREF
  TRANSACTID
  ZPLUS
  ZPLUSAGREF
  TABLEID
  ZCAV
  T1SCPTR
  ZPLUS
  ZPLUS
  ZONEFIELD
  ZUNIT
  ZSCIT1F1ID ;
EXIT;
T1SUBSPROC60)
IF RETRIEVEREF1AG = 1 GOTO T1SUBSPROC70 ;
IF RETRIEVEREF1AG = 2 GOTO T1SUBSPROC70 ;
IF RETRIEVEREF1AG = 3 GOTO T1SUBSPROC60 ;

IF RESULTCODE = #ZINVALIDCF THEN
  ESMES = #01NOOSPROVIDE ;
  GOTO SUBSPERR10 ;
FI;
IF RESULTCODE /= #ZSUCCESS THEN
  ESMES = #01NOOSPROVIDE ;
  GOTO SUBSPERR10 ;
FI;
OLDSCIT1 = TRRETRIEVAL1;
-----
                                >>>>>>>>>>
                                >RETRIEVEINF>
                                >>>>>>>>>>
RETRIEVEREF1AG = 3;
PROPERTY = #ZPROPERTY;
SEND RETRIEVEINF REFERENCE FSHREF WITH
  FSHREF
  TASCALUP
  CSUSCALUPREF
  TRANSACTID
  ZPLUS
  ZPLUSAGREF
  TABLEID
  ZCAV
  T1SCPTR
  ZPLUS
  ZPLUS
  ZONEFIELD
  ZUNIT
  ZSCPTR1F1ID;
EXIT;
T1SUBSPROC69)
IF RESULTCODE = #ZINVALIDCF THEN
  ESMES = #01NOOSPROVIDE ;
  GOTO SUBSPERR10 ;
FI;
IF RESULTCODE /= #ZSUCCESS THEN

```



```

IF @DISCITI (/= ZACTSC) THEN
    POSRES=ZEOSINVALID ;
    ROP = ;
    GOTO SUBSPROC120;
ELSE
    CPTR = T1SCPTR ;
    STATE = SUBSPROCEDURE;
    GOTO SUBSPROC70 ;
FI;

```

T1SUBSPROC75)

```

IF RESULTCODE = ZINVALDOKF THEN
    POSRES=ZEOSPROVIDE ;
    GOTO SUBSPERRIO ;
FI;

IF RESULTCODE (/= ZSUCCESS THEN
    POSRES=ZEOSCONG ;
    GOTO SUBSPERRIO ;
FI;

PROPERACC = TRETIEVALI ;

IF PROPERACC = 0 THEN
    POSRES=ZEOSPROVIDE ;
    GOTO SUBSPERRIO ;
FI;

```

ACTUALIZA CONTADORES

```

CASE FUNCODE IS
WHEN ZACTCWI / ZACTCAW DO
    BUSCOVIER = 1 ;
    COUNTERACT = COUNTERACT + 1;
    VALU = ZACT ;
    TOP = ;
    ROP = ;
WHEN ZDEACTCWI / ZDEACTCAW DO
    BUSCOVIER = 1 ;
    COUNTERDEACT = COUNTERDEACT + 1;
    VALU = ZDEACT ;
    TOP = ;
    ROP = ;
OTHERWISE DO
    POSRES=ZEOSINVALID;
    GOTO SUBSPERRIO ;
ESAC;

```

```

IF @DISCITI (/= ZACTSC) THEN
    POSRES=ZEOSINVALID ;
    ROP = ;
    GOTO SUBSPROC120;
ELSE
    CPTR = T1SCPTR ;
    STATE = SUBSPROCEDURE;
    GOTO SUBSPROC70 ;
FI;

```

RUTINA COMUN PARA AMBOS PROCEDIMIENTOS DE USUARIO

SUBSPROC50)

```
>>>>>>>>>>
>RETIEVALI>
>>>>>>>>>>
```

```

RETRIEVEREF = 1 ;
PROPERTY = ZUPPROPERTY ;
SEND RETRIEVALI REFERENCE FSHREF WITH
    SHPTR
    TASCAMP
    ZUSCAMPEF
    TRANACTIO
    ZOLVACDEF
    ZPROPERTTB
    ZPROPERTTB
    ZPROVIEW;
    CPTR
    PLUS
    PLUS
    PLUS
    ZONEFIELD
    ZUNIT
    PROPERTY ;

```

EXIT;
SUBSPROC60)

```

IF RESULTCODE = ZINVALDOKF THEN
    POSRES=ZEOSPROVIDE ;
    GOTO SUBSPERRIO ;
FI;

IF RESULTCODE (/= ZSUCCESS THEN
    POSRES=ZEOSCONG ;
    GOTO SUBSPERRIO ;
FI;

PROPERACC = TRETIEVALI ;

IF PROPERACC = 0 THEN

```

```

RETRIEVERETLAG = 2;
PROPERACC = ZACWPROPERTY;
SEMO RETRIEVEIN REFERENCE FSNREF WITH
      FSNREF
      FSNCALD
      SUSLAWEF
      TRANACTIO
      PLUS
      ZACWAGEF
      PROPERACC
      PROPERTIO
      CPTIR
      PLUS
      PLUS
      PLUSFIELD
      CUNIT
      PROPERTY;

```

```

EXIT;
FI;

```

ACTUALIZA CONTADORES

```

CASE FUNCODE IS
WHEN
  SUSCOUNTERP = ZACTCAW DO
  COUNTERACT = COUNTERACT + 1;
  VALUE = ZACT
  TOP
  POP
  WHEN ZACTCUI DO
  SUSCOUNTERP = COUNTERACT + 1;
  VALUE = ZACT
  TOP
  POP
  WHEN
  SUSCOUNTERP = 0; ZDEACTCAW DO
  COUNTERDEACT = COUNTERDEACT + 1;
  VALUE = ZDEACT
  TOP
  POP
  WHEN ZFACTCUI DO
  SUSCOUNTERP = 1;
  COUNTERDEACT = COUNTERDEACT + 1;
  VALUE = ZDEACT
  TOP
  POP
  OTHERWISE DO
  COSMESS = ZEOSINVALID;
  GOTO SUBSPERRIO;
ESAC;

```

```

IF OLDSCI /= ZACTSCI THEN
  COSMESS = ZEOSINVALID;
  POP
  GOTO SUBSPROC120;
FI;

```

SUBSPROC65)

```

IF RESULTCODE = ZINVALIDKF THEN
  COSMESS = ZEOSNOPROVIDE;
  GOTO SUBSPERRIO;
FI;
IF RESULTCODE /= ZSUCCESS THEN
  COSMESS = ZEOSNO;
  GOTO SUBSPERRIO;
FI;
PROPERACC = RETRIEVAL1;
IF PROPERACC = 0 THEN
  COSMESS = ZEOSNOPROVIDE;
  GOTO SUBSPERRIO;
FI;

```

```

CASE FUNCODE IS
WHEN
  SUSCOUNTERP = ZACTCAW DO
  COUNTERACT = COUNTERACT + 1;
  VALUE = ZACT
  TOP
  POP
  WHEN ZACTCUI DO
  SUSCOUNTERP = COUNTERACT + 1;
  VALUE = ZACT
  TOP
  POP
  WHEN
  SUSCOUNTERP = 0; ZDEACTCAW DO
  COUNTERDEACT = COUNTERDEACT + 1;
  VALUE = ZDEACT
  TOP
  POP
  WHEN ZFACTCUI DO
  SUSCOUNTERP = 1;
  COUNTERDEACT = COUNTERDEACT + 1;
  VALUE = ZDEACT
  TOP
  POP
  OTHERWISE DO

```



```

GOTO SETUP10 ;
??:
-----
SENDINREFFLAG = 1 ;
SEND SENDIN REFERENCE FSHREF WITH
    SMDR ;
    TASKCAMP ;
    CSUSCAMPREF,
    CLEGID ;
    MREFUNCODE ;
    ELEMENT7 ;
EXIT ;
INUSE40)
-----
CREATEREFFLAG = 1 ;
SEND CREATE REFERENCE FSHREF WITH
    SMDR ;
    TASKCAMP ;
    CSUSCAMPREF,
    MOCOT ;
    CLEGID ;
    NUMBERLENGTH,
    NUMBER1,
    NUMBER2,
    NUMBER3,
    NUMBER4,
    NUMBER5,
    NUMBER6,
    NUMBER7,
    NUMBER8,
    NUMBER9,
    NUMBER10,
    NUMBER11,
    NUMBER12,
    NUMBER13,
    NUMBER14,
    NUMBER15,
    NUMBER16,
    NUMBER17,
    NUMBER18,
    NUMBER19,
    NUMBER20 ;
EXIT ;
INUSE50)
    RESULTCODE = ZSUCCESS THEN
    ALEGID = CLEGID ;
    CALL ID = CALL ID ;
    GOTO READYFOREVENT ;
ELSE
    CLEGTFREE = TRUE ;
    CALL ZFAULT ;
    GOTO DISCONNECTION ;
FI ;
INUSE60)
-----
CASE EVENT IS
WHEN THROUGHCOMM DO
    ALEGID = TA00DATA1 ;
    CALL ID = TA00DATA2 ;
    ALEGMASK = ZCLEARED ;
    ALEGMMASK = ZCLEARED ;
    ALEGMASK = ZCLEARED ;
    ALEGMASK = ZCLEARED ;
ALEGMASK OFFHOOK = ZSET ;
ALEGMASK WINDOK = ZSET ;
STATE = WAITFOROFFHOOK ;
GOTO MONITOREVENT ;
WHEN ALGOSIC DO
    RESULTCODE = ZSUCCESS ;
    GOTO FINISH ;
WHEN THROUGH TBUSYSETUP, ZFAULTSETUP DO 1
    GOTO READYFOREVENT ;
WHEN FEATFUNCODE DO
    ALEGID = EVENTORID ;
    FEATUNCODE = TA00DATA1 ;
    GOTO INUSE70 ;
OTHERWISE DO
    ALEGTFREE = TRUE ;
    CLEGTFREE = TRUE ;
    CALL ZFAULT ;
    GOTO DISCONNECTION ;
ESAC ;
-----
INUSE70)
SENDINREFFLAG = 2 ;
SEND SENDIN REFERENCE FSHREF WITH
    SMDR ;
    TASKCAMP ;
    CSUSCAMPREF,
    CLEGID ;
    CALL IMAGE ;
    MREFUNCODE ;
    ELEMENT7 ;
    SMDR ;
EXIT ;
INUSE75)

```



```
GOTO DISCONNECTION;
E1:
ARCALLID = TRCEIVEINF1;
BLEGID = TRCEIVEINF3;
```

```
>>>>>>>>>>>>>>>>>>>>
>RECEIVEINF >
>>>>>>>>>>>>>>>>>>>>
```

```
RECEIVERETFLAG = 2;
SEND RECEIVEINF REFERENCE FSHREF WITH
    1:SHPR
    1:TASKAMP;
    CSUSCAMPREF;
    ACID;
    CLECID;
    2:PLUS;
    2:DATA;
    2:MESSTWO;
```

```
EXIT;
SETUP80)
MASK = VALD(MASK (0) #1000;
IF MASK = #0000 THEN
XLEGID = CLECID;
XCALLID = XCALLID;
CALLID = ARCALLID;
SETUP = 2:CALL;
GOTO FREERELASE;
E1;
CAVTYPEONE = TRCEIVEINF13;
IF CAVTYPEONE = 0 THEN
CAVTYPEONE = ZELEMENTS;
ELSE
CAVTYPEONE = ZELEMENT9;
E1;
```

```
CASE FEATNUM IS
WHEN ZCANNUMBER DO
IF FNUMCODE = ZCALLID GOTO SETUP130;
WHEN ZCVNUMBER DO
IF FNUMCODE = ZCALLID GOTO SETUP85;
OTHERWISE DO
GOTO EXIT;
ESAC;
```

```
SETUP85)
```

```
>>>>>>>>>>>>>>>>>>>>
>RECEIVEINF >
>>>>>>>>>>>>>>>>>>>>
```

```
RECEIVERETFLAG = 3;
SEND RECEIVEINF REFERENCE FSHREF WITH
    1:SHPR
    1:TASKAMP;
    CSUSCAMPREF;
    ACID;
    CLECID;
    2:ILITIES;
    2:MESSEONE;
```

```
EXIT;
SETUP100)
MASK = VALD(MASK (0) #0001;
IF MASK = #0000 THEN
XLEGID = CLECID;
XCALLID = ARCALLID;
CALLID = ARCALLID;
SETUP = 2:CALL;
GOTO FREERELASE;
E1;
BCALLTYPE = TRCEIVEINF1;
IF BCALLTYPE = 3
GOTO SETUP130;
```

```
SENDINERETFLAG = 2;
SEND SENDINF REFERENCE FSHREF WITH
    1:TASKAMP;
    CSUSCAMPREF;
    CLECID;
    2:INATED;
    2:MESSEONE;
    2:ZELEMENT17;
```

```
>>>>>>>>>>>>>>>>>>>>
>SENDINF >
>>>>>>>>>>>>>>>>>>>>
```

```
EXIT;
SETUP110)
IF RESULTCODE /= ZSUCCESS THEN
XLEGID = CLECID;
CALLID = ARCALLID;
XCALLID = XCALLID;
GOTO FREERELASE;
```

```

Fi;
                >>>>>>>>>>>>
                >>>>>>>>>>>>
SEND BELCONTROL REFERENCE FSHREF WITH
  FSHREF
  TASKCAMP
  CSUSCAWREF,
  ABCALLID;
EXIT;
SETUP120)
IF RESULTCODE /= ZSUCCESS THEN
  STATE = RELEASECONTROL ;
  RESULT = ZFAIL ;
  GOTO READYFOREVT;
Fi;

-----

CLEGROMASK = ZCLEARED ;
CLEGROMASK = ZCLEARED ;
CLEGROMASK = ZCLEARED ;
CLEGROMASK.ONHOOK = ZSET ;
STATE = RELEASE
GOTO MONITOREVT ;

-----

RELEASE10)
CASE TFEVNT IS
WHEN ZDHOOK = ZHSENDOD DO
  RESULT = ZSUCCESS ;
  EJECTFREE = ZTRUE ;
  GOTO DISCONNECTION ;
WHEN ZDISCOT ST DO
  RESULT = ZFAIL ;
  GOTO FINISH ;
OTHERWISE DO
  GOTO EXIT ;
ESAC ;

-----

ENVIO DE TONOS

SETUP130)
                >>>>>>>>>>>>
                >>>>>>>>>>>>
LOCKRESOURCEFLAG = 0 ;
SEND LOCKRESOURCES REFERENCE FSHREF WITH I
  FSHREF
  TASKCAMP
  CSUSCAWREF,
  RESIDTAG
  RESINFO ;
EXIT;
SETUP132)
RESOURCEID = TRESOURCEID ;
IF RESULTCODE /= ZSUCCESS THEN
  EJECTFREE = ZTRUE ;
  SENDINFRETFLAG = 0 ;
  RESULT = ZFAIL ;
  GOTO WAITFORRESUL10;
Fi;
RESOURCEFLAG = ZTRUE ;
                >>>>>>>>>>>>
                >>>>>>>>>>>>
SENDINFRETFLAG = 0 ;
SEND SENDINF REFERENCE FSHREF WITH
  FSHREF
  TASKCAMP
  CSUSCAWREF,
  ALREQID
  RESINFO,
  CAVITPTD,
  RESOURCEID ;
EXIT;
SETUP140)
IF RESULTCODE /= ZSUCCESS THEN
  REQID = CREQID ;
  CALLID = ABCALLID ;
  RESULT = ZFAIL ;
  GOTO FREERELEASE ;
Fi;

-----

                >>>>>>>>>>>>
                >>>>>>>>>>>>
                >>>>>>>>>>>>

```

```

ZENDINPREFLAG = 4
SEND SENDINREFERENCE FSHREF WITH
        SHPTR
        TASKCAMP
        COUNTCAMPREF,
        CLEVID
        CALLIMAGE,
        CALLINFO,
        CALLMAG,
        CALLMAGS,
        RESOURCEID ;

EXIT;
SETUP150;
IF RESULTCODE /= ZSUCCESS THEN
    CLEVID = CLEVID
    CALLID = ACALLID;
    CALLID = 2NULL;
    GOTO FREERELEASE ;
FI;
                                        |>>>>>>>>>>
                                        |>LOCKRESOURCES
                                        |>>>>>>>>>>

LOCKRESOURCEFLAG = 1;
SEND LOCKRESOURCES REFERENCE FSHREF WITH I
        SHPTR
        TASKCAMP
        COUNTCAMPREF,
        CLEVID
        CALLIMAGE
        CALLINFO ;

EXIT;
SETUP155;
IF RESULTCODE /= ZSUCCESS THEN
    CLEVID = CLEVID
    CALLID = ACALLID;
    CALLID = 2NULL;
    GOTO FREERELEASE ;

RESOURCEDEV = IRESOURCEID ;
RESOURCEDEVFLAG = ZTRUE ;

CASE FUNCODE IS
WHEN ZZPCAM = ZTRAFFIC DO
    ALEGNONMASK = ZCLEARED ;
    CLEGNONMASK = ZCLEARED ;
    ALEGNONMASK.ONHOOK = ZSET ;
    ALEGNONMASK.FLASH = ZSET ;
    BLEGNONMASK.ONHOOK = ZSET ;
    BLEGNONMASK.OFFHOOK = ZSET ;
    CLEGNONMASK.OFFHOOK = ZSET ;
    CLEGNONMASK.ONHOOK = ZSET ;
    END IS = CALLSTAT;
    GOTO MONITOREVENT ;

WHEN ZCWLID DO
    GOTO SETUP150;

OTHERWISE DO
    GOTO EXIT;
SETUP155;
SEND SENDINPREFLAG = 5
SEND SENDINREFERENCE FSHREF WITH
        SHPTR
        TASKCAMP
        COUNTCAMPREF,
        CLEVID
        CALLIMAGE,
        CALLINFO,
        CALLMAGS,
        CALLMAG ;

EXIT;
SETUP165;
RESOURCEBITFLAG = ZNULL; |> HANDOVER >
                                        |>>>>>>>>>>

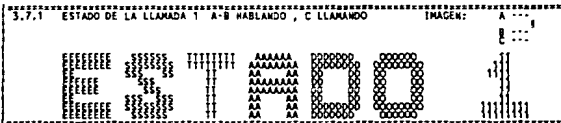
```



```

GOTO FREERELEASE10 ;
WHEN ZLEGDISC DO
  IF TEVENTOR/JID = ALECID THEN
    SELECTFREE = ZTRUE ;
    GOTO DISCONNECTION ;
  ELSE
    SELECTFREE = ZTRUE ;
    RESULT = ZSUCCESS ;
    GOTO DISCONNECTION ;
  FI ;
WHEN ZONHOOK DO
  SELECTFREE = ZTRUE ;
  RESULT = ZSUCCESS ;
  GOTO DISCONNECTION ;
  FI ;
SEND SPLIT REFERENCE FSHREF WITH
  FSHREF
  TASCAMP
  CASCAMPREF
  ALECID
  G
  RESOURCEID ;
EXIT ;
OTHERWISE DO
  GOTO READYTOEVENT ;
ES46)
WAITADFFHOOK10) GOTO DISCONNECTION ;
  >>>>>
  >SPLITT
  >>>>>

```



```

CSTATE1)
STATE = CALLSTATE1 ;
SUBACTION = WJID ;
RESOURCEFLAG = ZTRUE ;

CASE TEVENT IS
  WHEN ZFLASH DO
    IF TEVENTOR/JID = ALECID THEN
      RESOURCEFLAG = Z ;
      GOTO RELDEVICE ;
    FI ;
  WHEN ZONHOOK DO
    IF TEVENTOR/JID = ALECID
      GOTO CSTRONHOOK10 ;
    IF TEVENTOR/JID = BLECID THEN
      RESOURCEFLAG = Z ;
      GOTO CSTRONHOOK10 ;
    IF TEVENTOR/JID = CLECID THEN
      RESOURCEFLAG = Z ;
      GOTO RELDEVICE ;
    FI ;
  WHEN ZLEGDISC DO
    IF TEVENTOR/JID = ALECID THEN
      SELECTFREE = ZTRUE ;
      RESOURCEFLAG = Z ;
      GOTO RELDEVICE ;
    IF TEVENTOR/JID = BLECID THEN
      RESOURCEFLAG = Z ;
      GOTO RELDEVICE ;
    IF TEVENTOR/JID = CLECID THEN
      RESOURCEFLAG = Z ;
      GOTO RELDEVICE ;
    FI ;
  WHEN Z1 DO
    RESOURCEFLAG = Z ;
    GOTO RELDEVICE ;
  WHEN ZMESSENGER DO
    GOTO READYTOEVENT ;
  WHEN ZCALLIMAGEALTERED DO
    IF TADDDATA1 = ZREPLACED THEN
      IF TEVENTOR/JID = BLECID THEN
        CASE TADDDATA3 IS

```

- A CONJUNTA

- CUELGA

- A CUELGA
- B CUELGA

- C CUELGA

- A DESCONECTADO

- B DESCONECTADO

- C DESCONECTADO

- TIEMPO EXPIRADO

- FIN DE TONO

- IMAGEN ALTERADA

```

WHEN 0 2 DO
  A$CALL ID = TADDDATA2 ;
  A$CALL ID = TADDDATA4 ;
  GOTO READYFOREVENT ;
WHEN 1 2 DO
  A$CALL ID = TADDDATA2 ;
  A$CALL ID = TADDDATA4 ;
  GOTO READYFOREVENT ;
OTHERWISE DO
  EXIT ;
ELSE
  TADDDATA1 = CLEGRID THEN
  WHEN 0 2 DO
    A$CALL ID = TADDDATA2 ;
    A$CALL ID = TADDDATA4 ;
    GOTO READYFOREVENT ;
  WHEN 1 2 DO
    A$CALL ID = TADDDATA2 ;
    A$CALL ID = TADDDATA4 ;
    RELDEVCETAS = 5 ;
    GOTO RELDEVICE ;
  OTHERWISE DO
    EXIT ;
ELSE
  EXIT ;
ELSE
  TADDDATA = ZLEEDSCANONE THEN
  WHEN 0 2 DO
    RELDEVCETAS = 7 ;
    GOTO RELDEVICE ;
  OTHERWISE DO
    EXIT ;
ELSE
  EXIT ;
ELSE
  EXIT ;
FI ;

```

EVENTO IN ESPERADO

A CONMUTA ESTADO DE LA LLAMADA 1

```

CSTAFLASH10)
SUBSECTION = AFLASHES ;
SENDINRETFILAG = 0 ;
SEND SENDIN REFERENCE F$HREF WITH
  F$DIR
  TASKAMP*
  CSUSCAMREF*
  CLEEDATA*
  ZRESSTHREE ;
EXIT ;
CSTAFLASH20)
BRANCH ON SUBSECTION
TO CSTAFLASH30 IF AFLASHES
TO CSTAFLASH40 IF AMHOC
TO CSTAFLASH50 IF COMHOC
TO CSTAFLASH60 IF COISC
ELSE TO EXIT ;

```

A CONMUTA ESTADO DE LA LLAMADA 1

```

CSTAFLASH30)
IF SENDINRETFILAG = 2 GOTO CSTAFLASH50 ;
SENDINRETFILAG = 1 ;
SEND SENDIN REFERENCE F$HREF WITH
  F$DIR
  TASKAMP*
  CSUSCAMREF*
  CLEEDATA*
  ZRESSTHREE ;
EXIT ;
CSTAFLASH40)
SENDINRETFILAG = 2 ;
SEND SENDIN REFERENCE F$HREF WITH
  F$DIR
  TASKAMP*
  CSUSCAMREF*
  CLEEDATA*
  ZRESSTHREE ;
EXIT ;
CSTAFLASH50)
RELDEVCETAS = 12 ;
GOTO RELDEVICE10 ;
CSTAFLASH55)

```

```

SEND SPLIT REFERENCE FSHREF WITH
    FSNRFR
    TASCAMP'
    CSUSCAMPREF,
    ALECID ;
RESOURCEID;
EXIT;
CSIAFLASH0)
IF SUBSACTION = AONHOOK GOTO CSIAONHOOK20 ;

```

```

>>>>>
>SPLIT
>>>>>

```

```

JOINRETFLAG = 0;
SEND JOIN REFERENCE FSNREF WITH
    FSNRFR
    TASCAMP'
    CSUSCAMPREF,
    REFCALL ;
    ALECID ;
    REFCID ;
    RESOURCEID;
EXIT;
CSIAFLASH70)

```

```

>>>>>
>JOINR
>>>>>

```

```

IF SUBSACTION = BDISC GOTO CS1BDISC40 ;
IF JOINRETFLAG = 1 GOTO CSIAFLASH80;
IF RESULTCODE = 2SUCCESS THEN
    ACALLID = ACALLID ;
    GOTO CSIAFLASH90;
FI;

```

```

>>>>>
>JOINR
>>>>>

```

```

JOINRETFLAG = 1;
SEND JOIN REFERENCE FSNREF WITH
    FSNRFR
    TASCAMP'
    CSUSCAMPREF,
    REFCALL ;
    ALECID ;
    REFCID ;
    RESOURCEID;
EXIT;
CSIAFLASH80)

```

```

IF RESULTCODE = 2SUCCESS THEN
    ACALLID = ACALLID ;
    ACALLID = 2NULL ;
    CALLID = ACALLID ;
    RESULT = 2FREE ;
    GOTO FREERELEASE ;
ELSE
    RESULT = 2FAIL ;
    ALECIDOFREE = 2TRUE ;
    ALECIDOFREE = 2TRUE ;
    ALECIDOFREE = 2TRUE ;
    GOTO DISCONNECTION ;
FI;
CSIAFLASH90)

```

```

ALECONMARK = 2CLEAR;
ALECONMARK.ONHOOK = 2SET;
ALECONMARK.FLASH = 2SET;
ALECONMARK.ONHOOK = 2SET;

```

```

STATE = CALLSTATE2;
GOTO MONITOREVENT;

```

A CUELGA ESTADO DE LA LLAMADA ?

```

CSIAONHOOK0)
SUBSACTION = AONHOOK ;
SEND SPLIT REFERENCE FSNREF WITH
    FSNRFR
    TASCAMP'
    CSUSCAMPREF,
    ALECID ;
    REFCID ;
    RESOURCEID;
EXIT;
CSIAONHOOK20)

```

```

>>>>>>>>>
>FREE
>SPLIT
>>>>>>>>>

```

```

FREERETFLAG = 0;
SEND FREE REFERENCE FSNREF WITH

```

```

>>>>>>>>>
>>>>>>>>>

```



```

GOTO FREERELEASE ;
-----
B DESCONECTAED ESTADO DE LA LLAMADA1
CS1BD1SC10)
SUBSECTION = B01SC ;

SENDINFRFLAG = 0
SEND SENDINFR REFERENCE FSHREF WITH
    FSHREF
    TASKCALL
    CS/S/LMREF,
    CALLID,
    MESSAGE,
    MESSAGE,
    ELEMENTS ;

EXIT;
CS1BD1SC20)
IF SENDINFRFLAG = 1 GOTO CS1BD1SC30;
SENDINFRFLAG = 1
SEND SENDINFR REFERENCE FSHREF WITH
    FSHREF
    TASKCALL
    CS/S/LMREF,
    CALLID,
    MESSAGE,
    MESSAGE,
    ELEMENTS ;

EXIT;
CS1BD1SC30)
IF RESOURCETAG = 15;
CS1BD1SC35)
SEND JOIN REFERENCE FSHREF WITH
    FSHREF
    TASKCALL
    CS/S/LMREF,
    CALLID,
    RESOURCEID;

EXIT;
IF RESULTCODE = SUCCESS THEN
    ABCALLID = TDCALLID ;
    CALLID = ABCALLID;
    GOTO FREERELEASE ;
ELSE
    RESULT = ZFAIL ;
    ABCALLIDFREE = TRUE ;
    CALLIDFREE = TRUE ;
    GOTO DISCONNECTION ;
FI;

```

>>>>>
>JOINR
>>>>>

```

-----
C DESCONECTAED ESTADO DE LA LLAMADA1
CS1CD1SC10)
SUBSECTION = CD1SC;

SEND SENDINFR REFERENCE FSHREF WITH
    FSHREF
    TASKCALL
    CS/S/LMREF,
    CALLID,
    MESSAGE,
    MESSAGE,
    ELEMENTS ;

EXIT;
CS1CD1SC20)
IF RESOURCETAG = 17;
CS1CD1SC25)
CALLID = ABCALLID ;
CALLID = NULL ;
GOTO FREERELEASE ;

```

3.7.2 ESTADO DE LA LLAMADA 2 A-C HABLADO, S RETENIDO

TRAGEN: A -- 1

EEEEEE	SSSSSS	TTTTTTTT	AAAAAA	DDDDDD		727272
EEEEEE	SSSSSS	TTTTTTTT	AAAAAA	DDDDDD		222222
EEEE	SS	TT	AA	DD		22 22
EEEE	SS	TT	AA	DD		22 22
EEEEEE	SSSSSS	TTTTTTTT	AAAAAA	DDDDDD		222222
EEEEEE	SSSSSS	TTTTTTTT	AAAAAA	DDDDDD		222222

CS1ATE)
STATE = CALLSTATE2;
SUBSECTION = NONE;

<pre> CASE TEVENT IS WHEN ZFLASH DO IF TEVENTORJID = ALEGID GOTO CS2ZFLASH20; WHEN ZCALLIMAGEALTERED DO IF (ADDDATA) = ZLEGDISCAMOTOME THEN STATE = CALLALTER; FI; IF (ADDDATA) = ZREPLACEID GOTO ZREPLACEID; IF (ADDDATA) = ZLEGDISCAMOTOME GOTO CS2LEGCAMOTOME; WHEN ZONHOOK DO IF TEVENTORJID = ALEGID GOTO CS2ZONHOOK10; IF TEVENTORJID = BLEGID THEN KLEGID1 = BLEGID; CALLID = ACALLID; BCALLID = BCALLID; RESULT = ZSUCCESS; GOTO FREEPRELASE; FI; IF TEVENTORJID = CLEGID GOTO CS2CONHOOK10; WHEN ZLEGDISC DO IF TEVENTORJID = ALEGID THEN BLEGIDPRE = ZTRUE; KLEGIDPRE = ZTRUE; RESULT = ZFAIL; GOTO DISCONNECTION; FI; IF TEVENTORJID = BLEGID THEN CALLID ACALLID; BCALLID = BCALLID; RESULT = ZFAIL; GOTO FREERELEASE10; FI; IF TEVENTORJID = CLEGID THEN KLEGID2 = BLEGID; CALLID = ACALLID; RESULT = ZFAIL; GOTO JOINRELEASE; FI; OTHERWISE DO GOTO READYFOREVENT; ESAC EXIT; </pre>	<pre> ----- A COMUTA ----- IMAGEN ALTERADA ----- CUELGA ----- A CUELGA B CUELGA ----- C CUELGA ----- ALGUIEN FUE DESCONECTADO A DESCONECTADO ----- B DESCONECTADO ----- C DESCONECTADO ----- </pre>
<pre> ----- A COMUTA ESTADO DE LA LLAMADA 2 ----- CS2ZFLASH10) SUBSACTION = AFLASHES; SEMD SPLIT REFERENCE FSHREF WITH FSHREF; CSUSCAUP; CSUSCAUREF; ALEGID; EXIT; RESORCEID; CS2ZFLASH20) IF SUBSACTION = CONHOOK GOTO CS2CONHOOK20; IF SUBSACTION = COMHOOK GOTO CS2COMHOOK20; JOINRETFLAG = 0; SEMD JOIN REFERENCE FSHREF WITH FSHREF; CSUSCAUP; CSUSCAUREF; ZNEWCALL; ALEGID; BLEGID; RESORCEID; EXIT; CS2ZFLASH30) IF JOINRETFLAG = 1 GOTO CS2ZFLASH40; IF RESULTCODE = ZSUCCESS THEN ALEGID = TCALLID; GOTO CS2ZFLASH50; FI; JOINRETFLAG = 1; </pre>	<pre> >>>>>>>>>> > SPLIT > >>>>>>>>>> >>>>>>>> > JOINR >>>>>>>>>> >>>>>>>> > JOINR >>>>>>>>>> </pre>

```
SEND JOIN REFERENCE FSNREF WITH
    ASID,
    CSZASREF,
    CALL,
    ALECID,
    RESOURCEID;
```

EXIT;

```
CSZAF(LASW40)
IF RESULTCODE = ZSUCCESS THEN
```

```
    REPLICID = BLEGID;
    ACCALLID = ALECID;
    RECALLID = FSNREF;
    GOTO FREERELEASE;
ELSE
    RESULT = ZFAIL;
    REPLICID = BLEGID;
    ACCALLID = ALECID;
    RECALLID = FSNREF;
    GOTO DISCONNECTION;
```

```
CSZAF(LASW50);
```

ALECDOMASK = ZCLEARED;

CLECDOMASK = ZCLEARED;

ALECDOMASK.ONHOOK = ZSET;

ALECDOMASK.OFFHOOK = ZSET;

CLECDOMASK.ONHOOK = ZSET;

CLECDOMASK.OFFHOOK = ZSET;

STATE = CALLSTATE;

GOTO MONITOREVENT;

(MAGEN ALTERADA)

CSZRE(PLACE10)

REPLACLEG = TEVENTORJID;

REPLICID = TADDDATA;

NEWCALLID = TADDDATA;

IF REPLACLEG = CLEGID THEN

REPLICID = TADDDATA;

IF WORKSTATE = 1 THEN

ACCALLID = NEWCALLID;

GOTO CSZCONHOOKID;

IF WORKSTATE = 0 THEN

GOTO READYFORLEVER;

IF WORKSTATE = 2 THEN

ACCALLID = NEWCALLID;

STATE = CALLSTATE;

GOTO READYFORLEVER;

IF REPLACLEG = BLEGID THEN

REPLICID = TADDDATA;

IF WORKSTATE = 0 THEN

ACCALLID = NEWCALLID;

GOTO READYFORLEVER;

IF WORKSTATE IS

WHEN 1 DO

REPLICID = BLEGID;

ACCALLID = ALECID;

RECALLID = FSNREF;

GOTO FREERELEASE;

WHEN 2 DO

ACCALLID = NEWCALLID;

GOTO READYFORLEVER;

OTHERWISE DO

EXIT;

GOTO EXIT;

(MAGEN ALTERADA)

CSZLEGAND(ONE)

IF TADDDATA2 /= ALECID THEN

EXIT;

IF TEVENTORJID = BLEGID THEN

REPLICID = ALECID;

RECALLID = FSNREF;

GOTO FREERELEASE10;

IF TEVENTORJID /= CLEGID THEN

EXIT;

```

FI ;
                                >>>>>>>>>>>>
                                >LOCKRESOURCES
                                >>>>>>>>>>>>
LOCKRESOURCETAG = ?;
SEND LOCKRESOURCES REFERENCE FSHREF WITH 1
    FSHREF
    TASKCALLP
    CSUSCALLREF,
    ZRESINFO,
    ZRESINFO,
    ZRESINFO
EXIT;
CS2LEGANDTONE10)
    RESOURCEID = TRESOURCEID ;
IF ADDDATA3 = 0 THEN
    VALUE = ZRESINFO13 ;
ELSE ADDDATA3 = 1 THEN
    VALUE = ZELEMEN7 ;
ELSE
    EXIT ;
FI ;
                                >>>>>>>>>>>>
                                >SENDINF
                                >>>>>>>>>>>>
SENDINFREFLAG = 0 ;
SEND SENDINF REFERENCE FSHREF WITH
    FSHREF
    TASKCALLP
    CSUSCALLREF,
    ZRESINFO,
    ZRESINFO,
    ZRESINFO
EXIT;
CS2LEGANDTONE20)
    STATE = CALLSTATE13 ;
    GOTO READYFOREVENT ;

```

A CUELGA ESTADO DE LA LLAMADA 2

```

CS2AONHOOK10)
    SUBACTION = AONHOOK ;
                                >>>>>>>>>>>>
                                > SPLIT
                                >>>>>>>>>>>>
SEND SPLIT REFERENCE FSHREF WITH
    FSHREF
    TASKCALLP
    CSUSCALLREF,
    ZLEGID,
    RESOURCEID;
EXIT;
CS2AONHOOK20)
                                >>>>>>>>>>>>
                                >FREE
                                >>>>>>>>>>>>
SEND FREE REFERENCE FSHREF WITH
    FSHREF
    TASKCALLP
    CSUSCALLREF,
    ZLEGID ;
EXIT;
CS2AONHOOK30)
    IF SUBACTION = CONHOOK GOTO CS2CONHOOK30;
SEND SENDINF REFERENCE FSHREF WITH
    FSHREF
    TASKCALLP
    CSUSCALLREF,
    ZRESINFO,
    ZRESINFO13,
    ZRESINFO
EXIT;
CS2AONHOOK40)
IF RESULTCODE /= SUCCESS THEN
    ALEGIDFREE = TRUE ;
    RESULT = ZFAIL ;
    GOTO DISCONNECTION;
ELSE
    ALEGONMASK = ZCLEARED;
    CLEGONMASK = ZCLEARED;
    ALEGONMASK.OFFHOOK=ZSET;
    STATE = CALLSTATE7;

```

<pre> GOTO MONITOREVENT ; FI; C CUELGA ESTADO DE LA LLAMADA2 CS2CONHOOK10) SUBSACTION = CONHOOK ; SEM0 SPLIT REFERENCE FSHREF WITH FSHREF FASCCAMP FASCCUREF, ALEGID, RESOURCEID; EXIT; CS2CONHOOK20) SEM0 FREE REFERENCE FSHREF WITH FSHREF FASCCAMP FASCCUREF, CLEGID CLEGID = 2MULL ; EXIT; CS2CONHOOK30) ALEGID = ALEGID ; CLEGID = CLEGID ; RESULT = 2SUCCESS; GOTO JOINRELEASE; </pre>	<pre> >>>>>>>>>> > SPLIT > >>>>>>>>>> >>>>> >FREE >>>>> </pre>
<p>3.7.3 ESTADO DE LA LLAMADA 3 A-B HABLANDO , C RETENIDO IMAGEN: A --</p>	
<pre> EEEEEE 000000 000000 000000 000000 000000 000000 000000 000000 000000 EEEEEE 000000 000000 000000 000000 000000 000000 000000 000000 000000 EEEEEE 000000 000000 000000 000000 000000 000000 000000 000000 000000 EEEEEE 000000 000000 000000 000000 000000 000000 000000 000000 000000 EEEEEE 000000 000000 000000 000000 000000 000000 000000 000000 000000 EEEEEE 000000 000000 000000 000000 000000 000000 000000 000000 000000 EEEEEE 000000 000000 000000 000000 000000 000000 000000 000000 000000 EEEEEE 000000 000000 000000 000000 000000 000000 000000 000000 000000 EEEEEE 000000 000000 000000 000000 000000 000000 000000 000000 000000 EEEEEE 000000 000000 000000 000000 000000 000000 000000 000000 000000 </pre>	<pre> AAAAAA 000000 000000 000000 000000 000000 000000 000000 000000 000000 AAAAAA 000000 000000 000000 000000 000000 000000 000000 000000 000000 AAAAAA 000000 000000 000000 000000 000000 000000 000000 000000 000000 AAAAAA 000000 000000 000000 000000 000000 000000 000000 000000 000000 AAAAAA 000000 000000 000000 000000 000000 000000 000000 000000 000000 AAAAAA 000000 000000 000000 000000 000000 000000 000000 000000 000000 AAAAAA 000000 000000 000000 000000 000000 000000 000000 000000 000000 AAAAAA 000000 000000 000000 000000 000000 000000 000000 000000 000000 AAAAAA 000000 000000 000000 000000 000000 000000 000000 000000 000000 AAAAAA 000000 000000 000000 000000 000000 000000 000000 000000 000000 </pre>
<pre> CSTATE) STATE = CALLSTATE; SUBSACTION = NONE; CASE TEVENT IS WHEN ZFLASH DO IF TEVENTOBJID = ALEGID GOTO CS2AFLASH; WHEN ZCALLIMAGALTERED DO CLEGID = ALEGID; BLEGID = BLEGID; CALLID = 2MULL; GOTO CSTATE; WHEN ZCONHOOK DO IF TEVENTOBJID = ALEGID GOTO CS2CONHOOK10; IF TEVENTOBJID = BLEGID THEN ALEGONMASK = ZCLEARED; CLEGONMASK = ZCLEARED; ALEGONMASK_CONHOOK = ZSET; ALEGONMASK_FLASH = ZSET; CLEGONMASK_CONHOOK = ZSET; CLEGONMASK_FLASH = ZSET; STATE = CALLSTATE; GOTO MONITOREVENT; IF TEVENTOBJID = CLEGID THEN ZTEGID = ALEGID; ZCLEGID = BLEGID; ACCALLID = 2MULL; RESULT = 2SUCCESS; GOTO FREERELEASE; FI; WHEN ZLEGDISC DO IF TEVENTOBJID = ALEGID THEN ALEGIFREE = 2TRUE; RESULT = 2FALSE; GOTO DISCONNECTION; IF TEVENTOBJID = BLEGID THEN ZTEGID = ALEGID; ZCLEGID = CLEGID; GOTO JOINRELEASE; IF CLEGID = TEVENTOBJID THEN RESULT = 2FALSE; CALLID = ABCALLID; </pre>	<pre> A CONMUTA A CUELGA IMAGEN ALTERADA CUELGA A CUELGA C CUELGA ALGUIEN COLGO A DESCONECTADO B DESCONECTADO C DESCONECTADO </pre>

```

ACALLID = ZMALL;
GOTO FREERELEASED;
FI;
    OTHERWISE DO
        GOTO READYFOREVENT;
    ESAC;
    EXIT;
    A DESCONECTADO
-----
A COMIENZA ESTADO DE LA LLAMADA 3
CS3AFLASH10)
    SUBSACTION = AFLASHES ;
    SEND SPLIT REFERENCE FSHREF WITH
        FSHREF;
        TASKCAND;
        CSUSCAWREF;
        ALEGID;
        RESOURCEID;
    EXIT;
CS3AFLASH20)
    IF SUBSACTION = AONHOOK GOTO CS3AONHOOK20;
    JOINRETFLAG = 0;
    SEND JOIN REFERENCE FSHREF WITH
        FSHREF;
        TASKCAND;
        CSUSCAWREF;
        HEMCALL;
        PLUS;
        ALEGID;
        CLEGID;
        RESOURCEID;
    EXIT;
CS3AFLASH30)
    IF JOINRETFLAG = 1 GOTO CS3AFLASH40;
    IF RESULTCODE = ZSUCCESS THEN
        ACALLID = TCALLID;
        GOTO CS3AFLASH50;
    FI;
    JOINRETFLAG = 1;
    SEND JOIN REFERENCE FSHREF WITH
        FSHREF;
        TASKCAND;
        CSUSCAWREF;
        HEMCALL;
        PLUS;
        ALEGID;
        CLEGID;
        RESOURCEID;
    EXIT;
CS3AFLASH40)
    IF RESULTCODE = ZSUCCESS THEN
        ALEGID = CLEGID;
        ACALLID = TCALLID;
        CCALLID = ACALLID;
        RESULT = ZSUCCESS;
        GOTO FREERELEASE;
    ELSE
        RESULT = ZFAIL;
        ALEGIDFREE = ZTRUE;
        CLEGIDFREE = ZTRUE;
        GOTO DISCONNECTION;
    FI;
CS3AFLASH50)
    ALEGONMASK = ZCLEARED;
    CLEGONMASK = ZCLEARED;
    ALEGONMASK.ONHOOK = ZSET;
    ALEGONMASK.FLASH = ZSET;
    CLEGONMASK.ONHOOK = ZSET;
    STATE = CALLSTATEZ;
    GOTO MONITOREVENT;
-----
A CUELGA ESTADO DE LA LLAMADA 3
CS3AONHOOK10)
    SUBSACTION = AONHOOK ;
    SEND SPLIT REFERENCE FSHREF WITH
        FSHREF;
        TASKCAND;
        CSUSCAWREF;

```



```

        ALEGID ,
        RE$OURCEID;
EXIT;
CS$AONHOOK(20)
        >>>>>>
        >>>>>>
        >>>>>>
SEND FREE REFERENCE F$HREF WITH
        F$HREF
        T$ASCCAMP
        C$DISCAMREF,
        BLEGID
        BLEGID = ZNULL;
EXIT;
CS$AONHOOK(30)
SEND SENDING REFERENCE F$HREF WITH
        F$HREF
        T$ASCCAMP
        C$DISCAMREF,
        Z$E$C$INFO,
        Z$E$S$ONH,
        Z$E$ENTID,
        Z$E$TAG
        RESOURCEID ;
EXIT;
CS$AONHOOK(40)
IF RESULTCODE /= Z$SUCCESS THEN
        GOTO ZFREE = TRUE;
        RESULT = ZFAIL;
        GOTO DISCONNECTION;
ELSE
        ALEGMONM$K = ZCLEARED;
        BLEGMONM$K = ZCLEARED;
        CLEGMONM$K = ZCLEARED;
        ALEGMONM$K.OFFHOOK = ZSET;
        BLEGMONM$K.OFFHOOK = ZSET;
        CLEGMONM$K.OFFHOOK = ZSET;
        STATE = CALLSTATE1;
        GOTO MONITOREVENT;
FI;
    
```

3.7.4 ESTADO DE LA LLAMADA 4 A-B HABLANDO C LLAMANDO IMAGEN: A B
 B SUPERVISION DE TIEMPO

```

EEEEEEEE SSSSSSS TTTTTTT AAAAAA 000000 000000 444444
EEEEEEEE SSSSSSS SSSSSSS AAAA AAAA AAAA AAAA
EEEEEEEE SSSSSSS SSSSSSS AAAA AAAA AAAA AAAA
EEEEEEEE SSSSSSS SSSSSSS AAAA AAAA AAAA AAAA
EEEEEEEE SSSSSSS SSSSSSS AA AA AA AA
EEEEEEEE SSSSSSS SSSSSSS AA AA AA AA
    
```

```

C$STATE;
STATE = CALLSTATE4;
SUBACTION = NONE;

CASE TEVENT IS
WHEN ZFLASH DO
    IF TEVENTORJID = ALEGID
        GOTO CS$ZFLASHID;
WHEN ZONHOOK DO
    IF TEVENTORJID = ALEGID THEN
        GOTO CS$AONHOOK;
    IF TEVENTORJID = BLEGID
        GOTO CS$BONHOOK;
WHEN ZOFFHOOK DO
    IF TEVENTORJID = BLEGID THEN
        ALEGMONM$K = ZCLEARED;
        BLEGMONM$K = ZCLEARED;
        CLEGMONM$K = ZCLEARED;
        ALEGMONM$K.OFFHOOK = ZSET;
        BLEGMONM$K.OFFHOOK = ZSET;
        CLEGMONM$K.OFFHOOK = ZSET;
        STATE = CALLSTATE1;
        GOTO MONITOREVENT;
    FI;
WHEN ZLEGDISC DO
    IF TEVENTORJID = ALEGID THEN
        RELRESOURCEID = Z0;
        STATE = CALLSTATE1;
        BLEGIFREE = TRUE;
        RESULT = ZFAIL;
        GOTO RELEDEVICEID;
    FI;
    
```

A CONMUTA

CUELGA

A CUELGA
 C CUELGA

ALGUIEN COLGO

A DESCONECTADO

```

IF TEVENTORID = BLEGID THEN
  GOTO CS180152(1);
;;
TEVENTORID = BLEGID
GOTO CS180152(1);
WHEN ZMESSAGEDO DO
  GOTO READTOEVENT;

WHEN ZCALLIMAGEALTERED DO
  IF TADDDATA1 = ZREPLACEID THEN
    IF TEVENTORID = BLEGID THEN
      CASE TADDDATA1 IS
        WHEN 0 DO
          BLEGID = TADDDATA2;
          BLEGID = TADDDATA2;
          GOTO READTOEVENT;
        WHEN 1 DO
          BLEGID = TADDDATA2;
          BLEGID = TADDDATA2;
          GOTO CS180152(1);
        OTHERWISE DO
          EXIT;
        ESAC;
      ELSE AC;
    ELSE AC;
    ELSE AC;
    ELSE AC;
    ELSE AC;
    ELSE AC;
  OTHERWISE DO
    GOTO EXIT;
ESAC;
EXIT;

```

B DESCONECTADO

C DESCONECTADO

IMAGEN ALTERADA

A CONMUTA ESTADO DE LA LLAMADA 4

```

CS4AF1ASH10)
SUBSACTON = AFLASHES;

SEMDINFRFLAG = 0;
SEMD SENDINFR REFERENCE FSHREF WITH
  FSHPTR
  TASKAMP'
  CSUSCAMREF,
  BLEGID
  TADDDATA1,
  ZMESSAGE,
  ZELEMENTS;

EXIT;

CS4AF1ASH20)
BRANCH ON SUBSACTON
  TO CS4AF1ASH10 IF AFLASHES
  TO CS180152(2) IF CONHOOK
  TO CS40152(2) IF CDISC
  ELSE TO EXIT;

CS4AF1ASH30)
IF SEMDINFRFLAG = 1 GOTO CS4AF1ASH35;
IF SEMDINFRFLAG = 2 GOTO CS4AF1ASH40;

SEMDINFRFLAG = 1;
SEMD SENDINFR REFERENCE FSHREF WITH
  FSHPTR
  TASKAMP'
  CSUSCAMREF,
  BLEGID
  TADDDATA1,
  ZMESSAGE,
  ZELEMENTS;

EXIT;

CS4AF1ASH35)
SEMDINFRFLAG = 2;
SEMD SENDINFR REFERENCE FSHREF WITH
  FSHPTR
  TASKAMP'
  CSUSCAMREF,
  BLEGID
  TADDDATA1,
  ZMESSAGE,
  ZELEMENTS;

EXIT;

CS4AF1ASH40)
TEVENTORID = 18;
GOTO RELDEVICE10;

CS4AF1ASH45)

```

A CONMUTA ESTADO DE LA LLAMADA 1

```

|>>>>>
|> SPLIT
|>>>>>

```

```

SEMD SPLIT REFERENCE FSHREF WITH
  FSHPTR
  TASKAMP'
  CSUSCAMREF,
  BLEGID
  TADDDATA1,
  ZMESSAGE,
  ZELEMENTS;

```

```

RESOURCEID;
EXIT;
CS4AFLASH50)

SEND FREE REFERENCE FSHREF WITH
    FSHPTR
    TASKCAMP
    CSUSCAUREF,
    ALEGID
    CLEGID = ZNULL ;
EXIT;
CS4AFLASH60)
IF SUBSACTION = CONHOOK GOTO CS4CONHOOK20;
    >>>>>
    >>>>>
    >>>>>
    >FREE
    >JOIN
    >>>>>>>>>>
    >>>>>>>>>>

JOINRETFAG = D;
SEND JOIN REFERENCE FSHREF WITH
    FSHPTR
    TASKCAMP
    CSUSCAUREF,
    NEWCALL
    FSHREF
    ALEGID
    CLEGID;
RESOURCEID;
EXIT;
CS4AFLASH70)

IF RESULTCODE = ZSUCCESS THEN
    ACALLID = ACALLID
    CALLID = ACALLID
    ACALLID = ZNULL
    RESULT = ZFAIL
    GOTO FREERELEASE10 ;
ELSE
    RESULT = ZFAIL
    ALEGIDFREE = ZTRUE
    GOTO DISCONNECTION ;
FI;
    
```

C DUELA ESTADO DE LA LLAMADA 4

```

CS4CONHOOK10)
SUBSACTION = CONHOOK ;
SEND SENDINF REFERENCE FSHREF WITH
    FSHPTR
    TASKCAMP
    CSUSCAUREF,
    ALEGID
    FSHINFO
    MESSAGE
    ZELEMENT1 ;
EXIT;
CS4CONHOOK20)
RELRESOURCEID = 19;
GOTO RELDEVICE10;
CS4CONHOOK25)

SEND FREE REFERENCE FSHREF WITH
    FSHPTR
    TASKCAMP
    CSUSCAUREF,
    CLEGID
    CLEGID = ZNULL ;
EXIT;
CS4CONHOOK30)
ALEGONMASK = ZCLEARED;
CLEGONMASK = ZCLEARED;
ALEGONMASK.ONHOOK = ZSET;
STATE = CALLSTATE;
GOTO MONITOREVENT;
    
```

C DESCONECTADO ESTADO 4

```

CS4CDISC10)
SUBSACTION = CRIFC.
SEND SENDINF REFERENCE FSHREF WITH
    FSHPTR
    TASKCAMP
    CSUSCAUREF,
    ALEGID
    FSHINFO
    MESSAGE
    ZELEMENT1 ;
EXIT;
CS4CDISC20)
RELRESOURCEID = 21;
GOTO RELDEVICE10;
CS4CDISC25)
ALEGONMASK = ZCLEARED;
CLEGONMASK = ZCLEARED;
ALEGONMASK.ONHOOK = ZSET;
    
```



```

    OTHERWISE DO
        GOTO READYFOREVENT ;
    ESAC
    EXIT;
A CONMUTA ESTADO DE LA LLAMADA 5
CSSAFLASH10)
    SUBSACTION = AFLASHES;
    SEND SPLIT REFERENCE FSHREF WITH
        FSHPTR
        TASKCALP
        CSUSCALREF
        ALEGID
        0
        RESOURCEID;
    EXIT;
CSSAFLASH20)
    SEND FREE REFERENCE FSHREF WITH
        FSHPTR
        TASKCALP
        CSUSCALREF
        BLEGID
        ZNULL ;
    EXIT;
CSSAFLASH30)
    IF SUBSACTION = CONHOOK GOTO CSSCONHOOK20;
    XLEGID1 = ALEGID;
    XLEGID2 = BLEGID;
    RESULT = ZSUCCESS ;
    GOTO JOINRELEASE;

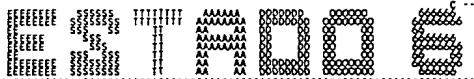
```

```

C CUELGA ESTADO DE LA LLAMADA 5
CSSCONHOOK10)
    SUBSACTION = CONHOOK ;
    SEND FREE REFERENCE FSHREF WITH
        FSHPTR
        TASKCALP
        CSUSCALREF
        CLEGID
        ZNULL ;
    EXIT;
CSSCONHOOK20)
    ALEGNONMASK = ZCLEARED;
    ALEGNONMASK = ZCLEARED;
    ALEGNONMASK_ONHOOK = ZSET;
    STATE = CALLSTATED;
    GOTO MONITOREVENT;

```

3.7.6 ESTADO DE LA LLAMADA 6 A RELAMADO POR C, B DESC. IMAGEN: A ---



```

CSTATE6)
    STATE = CALLSTATE6;
    SUBSACTION = NONE;
CASE TEVENT IS
    WHEN ZONHOOK DO
        IF TEVENTOBJID = CLEGID
            GOTO CSSCONHOOK10;
    WHEN ZMESSENGED DO
        RESULT = ZFAIL ;
        ALEGTOPFREE = ZTRUE ;
        ALEGTOPFREE = ZTRUE ;
        GOTO DISCONNECTION;
    WHEN TOPHOOK DO
        IF TEVENTOBJID = ALEGID THEN
            XLEGID1 = ALEGID;
            XLEGID2 = CLEGID;
            RESULT = ZSUCCESS;

```

```

GOTO JOINRELEASE ;
FI ;
WHEN ZLEGDISC DO
    IF TEVENTORJID = ALEGID THEN
        ALEGTOFREE = ZTRUE ;
        GOTO DISCONNECTION ;
    FI ;
    IF TEVENTORJID = CLEGID
        GOTO CS6CDISC5 ;
    OTHERWISE DO
        GOTO EXIT ;
EXIT ;

```

ALGUIEN COLGO
A DESCONECTADO

C DESCONECTADO

```

C CUELGA ESTADO DE LA LLAMADA 6
CS6CDHOOK(0)
SEND SENDIN REFERENCE FSHREF WITH
    CSHPTR
    TASKCALD
    CSUSCAUREF
    PLANINFO
    ZELEMENT1 ;
EXIT ;
CS6CDHOOK(0)
SEND SENDIN REFERENCE FSHREF WITH
    CSHPTR
    TASKCALD
    CSUSCAUREF
    PLANINFO
    ZELEMENT1 ;
EXIT ;
CS6CDHOOK(0)
SEND SENDIN REFERENCE FSHREF WITH
    CSHPTR
    TASKCALD
    CSUSCAUREF
    PLANINFO
    ZELEMENT1 ;
EXIT ;
CS6CDISC10)
ALEGTOFREE = ZTRUE ;
RESULT = ZFAIL ;
GOTO DISCONNECTION ;

```

```

C DESCONECTADO ESTADO DE LA LLAMADA 6
CS6CDISC5)
SEND SENDIN REFERENCE FSHREF WITH
    CSHPTR
    TASKCALD
    CSUSCAUREF
    PLANINFO
    ZELEMENT1 ;
EXIT ;
CS6CDISC10)
ALEGTOFREE = ZTRUE ;
RESULT = ZFAIL ;
GOTO DISCONNECTION ;

```

3.7.7 ESTADO DE LA LLAMADA 7 A RETROLLAMADO POR B, C DISC. IMAGEN: A ---



```

CSTATE(7)
STATE = CALLSTATE7 ;
SUBSACTION = NONE ;

CASE TEVENT IS
WHEN ZONHOOK DO
    IF TEVENTORJID = BLEGID
        GOTO CS7BONHOOK0 ;
WHEN ZRESSENDID DO
    RESULT = ZFAIL ;
    ALEGTOFREE = ZTRUE ;
    GOTO DISCONNECTION ;
WHEN ZOFFHOOK DO
    IF TEVENTORJID = ALEGID THEN
        ALEGTOFREE = ZTRUE ;
        RESULT = ZFAIL ;
        GOTO JOINRELEASE ;
    FI ;
    IF TEVENTORJID = BLEGID
        GOTO READYFOREVER ;
WHEN ZLEGDISC DO

```

CUELGA

C CUELGA

DESCONECTADO
A DESCONECTADO

ALGUIEN COLGO

```

IF TEVENTOBJID = BLEGID
GOTO CS7B01SC05
IF TEVENTOBJID = ALEGID THEN
ALEGIOFREE = TRUE;
RESULT = 27A1;
GOTO DISCONNECTION;
FI;
OTHERWISE DO
GOTO EXIT;
ESAC;
EXIT;

```

A DESCONECTADO
B DESCONECTADO

B CUELGA ESTADO DE LA LLAMADA 7

```

CS7B0NHOOK10)
SEMOINFRETFLAG = 0
SEMO SENDIN REFERENCE FSHREF WITH
FSMPTA
TASCALWD
TASCALWD
TASCALWD
ALEGIO
TASCALWD
TASCALWD
TASCALWD
ELEMENT1;
EXIT;

```

```

CS7B0NHOOK20)
IF SEMOINFRETFLAG = 1 GOTO CS7B01SC10;
ALEGIOFREE = TRUE;
RESULT = 27A1;
GOTO DISCONNECTION;

```

B ON DISC ESTADO DE LA LLAMADA 7

```

CS7B01SC05)
SEMOINFRETFLAG = 1
SEMO SENDIN REFERENCE FSHREF WITH
FSMPTA
TASCALWD
TASCALWD
TASCALWD
ALEGIO
TASCALWD
TASCALWD
TASCALWD
ELEMENT1;
EXIT;

```

```

CS7B01SC10)
ALEGIOFREE = TRUE;
RESULT = 27A1;
GOTO DISCONNECTION;

```

3.7.8 ESTADO DE LA LLAMADA A A B HABLANDO, B SUPERVISION DE TIEMPO, IMAGEN: A ---

```

EEEEEEEE 00000000 11111111  AAAAAA  DDDDDDD  OOOOOO  B - T
EEEEEEEE 00000000 11111111  AAAAAAAA  DDDDDDD  OOOOOO  B - T
EEEEEEEE 00000000 11111111  AA  AA  DD  DD  OO  OO  B - T
EEEEEEEE 00000000 11111111  AAAAAAAA  DD  DD  OO  OO  B - T
EEEEEEEE 00000000 11111111  AAAAAAAA  DD  DD  OO  OO  B - T
EEEEEEEE 00000000 11111111  AA  AA  DD  DD  OO  OO  B - T
EEEEEEEE 00000000 11111111  AA  AA  DD  DD  OO  OO  B - T
EEEEEEEE 00000000 11111111  AA  AA  DD  DD  OO  OO  B - T

```

```

CSTATEB)
STATE = CALLSTATEB;
SUBSACTION = NONE;

```

```

CASE TEVENT IS
WHEN ZONHOOK DO
IF TEVENTOBJID = ALEGID
GOTO CSB0NHOOK0;
WHEN ZOFFHOOK DO
IF TEVENTOBJID = BLEGID THEN
CALLID = ABCALLID;
ACALLID = 27A1;
GOTO FREERELEASETO;
FI;
WHEN ZLEGDISC DO
IF TEVENTOBJID = ALEGID THEN
ALEGIOFREE = TRUE;
RESULT = 27A1;
GOTO DISCONNECTION;
FI;
IF TEVENTOBJID = BLEGID THEN
ALEGIOFREE = TRUE;
RESULT = 27A1;
GOTO DISCONNECTION;
OTHERWISE DO
GOTO EXIT;
ESAC;

```

CUELGA
A CUELGA
B CONTESTA
ALGUIEN COLGO
A DESCONECTADO
B DESCONECTADO

EXIT;

A CUELGA ESTADO DE LA LLAMADA B

CSBAOHK010)

SEND SPLIT REFERENCE FS#REF WITH

```

FS#PID
TASKCALL
CS#CALLREF
ALEGID
Z
RESOURCEID;
    
```

EXIT;

CSBAOHK020)

```

ALCO#FREE = ZTRUE;
RESUL#FREE = ZTRUE;
GOTO DISCONNECTION;
    
```

```

>>>>>>>>>>
> SPLIT >
>>>>>>>>>>
    
```

3.7.9 ESTADO DE LA LLAMADA 9 ESTABLECIMIENTO DE LA RETROLAMDA HACIA A

IMAGEN: C

```

EEEEEEEE SSSSSSS TTTTTTT AAAAAA DDDDDD OOOOOO
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
    
```

CSTATE0)

```

STATE = CALLSTATE0;
SUBACTION = NONE;
CASE TEVENT IS
    
```

```

WHEN ZOHK00K DO
IF TEVENTORID = CLEGID THEN
RESULT = ZSUCCESS;
GOTO DISCONNECTION;
    
```

```

WHEN ZLEGDISC DO
IF TEVENTORID = CLEGID THEN
RESULT = ZSUCCESS;
GOTO FINISH;
    
```

```

WHEN ZOHK0SETUP ZBUSYSETUP ,
ZFAULTSETUP DO
    
```

```

ALCO#ONMAX = ZCLEARED;
CLE#ONMAX = ZCLEARED;
CLE#ONMAX.D#HOOK = ZSET;
STATE = CALLSTATE1;
GOTO MONITOREVENT;
    
```

WHEN ZTHROUGHCONN DO

```

ALEGID = ZADDDATA1;
ACALLID = ZADDDATA2;
ALCO#ONMAX = ZCLEARED;
CLE#ONMAX = ZCLEARED;
CLE#ONMAX.D#HOOK = ZSET;
STATE = CALLSTATE0;
GOTO MONITOREVENT;
    
```

OTHERWISE DO

GOTO EXIT;

ESAC

EXIT;

CUELGA

C CUELGA

ALGUIEN COLGO
C DESCONECTADO

3.7.10 ESTADO DE LA LLAMADA 10 A LLAMANDO, C LLAMANDO

IMAGEN: A

```

EEEEEEEE SSSSSSS TTTTTTT AAAAAA DDDDDD OOOOOO
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
EEEEEEEE SSSSSSS TTTTTTT AAAAAA AA
    
```

CSTATE10)

```

STATE = CALLSTATE10;
SUBACTION = NONE;
CASE TEVENT IS
    
```

```

WHEN ZOFFHOOK DO
IF TEVENTORID = ALEGID THEN
CALLID = ACALLID;
ACALLID = ZNULL;
RESULT = ZSUCCESS;
GOTO FREERELASETO;
    
```

DESCUELGA

A CONTESTA


```

FI:
WHEN ZONHOOK DO
IF TEVENTORJID =CLEGID
GOTO ZESTOONHOOKS ;
WHEN ZLEGDISC DO
IF TEVENTORJID =CLEGID THEN
ALEGTOFREE = ZTRUE ;
RESULT = ZSUCCESS ;
GOTO FINISH ;
OTHERWISE DO
GOTO EXIT ;
ESAC ;
EXIT ;

```

```

CUELGA
-----
ALGUIEN COLGO
-----

```

C510CONHOOKOS)

```

SPLITRETFLAG = 0 ;
SEND SPLIT REFERENCE FSHREF WITH
        FSHREF
        TASKCAMP
        CUSCAUREF,
        CLEGID,
        RESOURCEID;
EXIT;
C510CONHOOKID)
ALEGTOFREE = ZTRUE;
RESULT = ZSUCCESS;
GOTO DISCONNECT ;

```

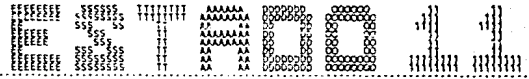
```

>>>>>>>>>
>> SPLIT >
>>>>>>>>>
C CUELGA

```

3.7.11 ESTADO DE LA LLAMADA 11 A OCUPADO/CONGESTIONADO C ESCUCHA TONO DE OCUPADO

IMAGEN: C ---



CSTATE11)

```

STATE = CALLSTATE11;
SUBSECTION = NONE ;
CASE TEVENT IS
WHEN ZONHOOK DO
IF TEVENTORJID =CLEGID THEN
ALEGTOFREE = ZTRUE ;
RESULT = ZSUCCESS ;
GOTO DISCONNECT ;
WHEN ZLEGDISC DO
IF TEVENTORJID =CLEGID THEN
RESULT = ZSUCCESS ;
GOTO FINISH ;
OTHERWISE DO
GOTO EXIT ;
ESAC ;
EXIT ;

```

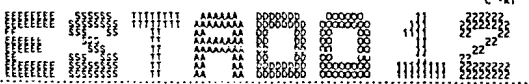
```

CUELGA
-----
C CUELGA
-----
ALGUIEN COLGO
-----
C DESCONECTADO
-----

```

3.7.12 ESTADO DE LA LLAMADA 12 A LLLAMANDO C LLAMANDO B RETENIDO MAGEN:

M ---
H ---
RT ---



CSTATE12)

```

STATE = CALLSTATE12;
SUBSECTION = NONE ;
CASE TEVENT IS
WHEN ZFLASH DO
IF TEVENTORJID =ALEGID
GOTO C512AFASH ;
WHEN ZONHOOK DO
IF TEVENTORJID =ALEGID
GOTO C512ONHOOK ;
IF TEVENTORJID =BLEGID
GOTO C512BONHOOK ;

```

```

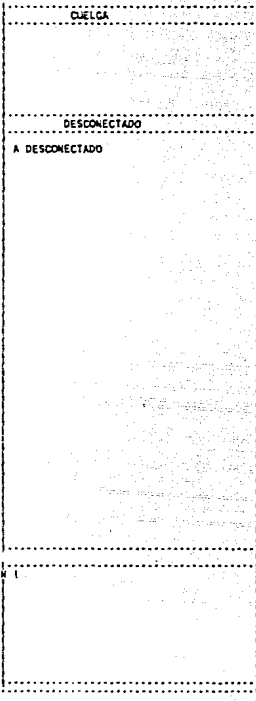
A CONMUTA
-----
CUELGA
-----
A CUELGA
-----
B CUELGA
-----

```

```

WHEN ZOFFHOOK DO
    IF TEVENTORJID = CLEGID THEN
        ALEGMONMASK = ZCLEARED;
        CLEGMONMASK = ZCLEARED;
        ALEGMONMASK.ONHOOK = ZSET;
        CLEGMONMASK.ONHOOK = ZSET;
        STATE = CALLSTATE7;
        GOTO MONITOREVENT;
    FI;
WHEN ZLEGDISC DO
    IF TEVENTORJID = ALEGID THEN
        BLEGIFREE = TRUE;
        CLEGMONMASK = ZCLEARED;
        RESULT = ZFAIL;
        GOTO DISCONNECTION;
    FI;
    IF TEVENTORJID = BLEGID THEN
        ALEGMONMASK = ZCLEARED;
        CLEGMONMASK = ZCLEARED;
        ALEGMONMASK.ONHOOK = ZSET;
        CLEGMONMASK.OFFHOOK = ZSET;
        STATE = CALLSTATE14;
        GOTO MONITOREVENT;
    FI;
    IF TEVENTORJID = CLEGID THEN
        ALEGMONMASK = ZCLEARED;
        CLEGMONMASK = ZCLEARED;
        ALEGMONMASK.OFFHOOK = ZSET;
        STATE = CALLSTATE7;
        GOTO MONITOREVENT;
    FI;
WHEN ZCALLIMAGEALTERED DO
    IF TADDDATA1 = ZREPLACED THEN
        REPLACLEG = TEVENTORJID;
        HOOKSTATE = TASKSTATE;
        NEWCALLID = TADDDATA2;
        IF REPLACLEG = BLEGID THEN
            BLEGIFREE = 0;
            ABCALLID = NEWCALLID;
            GOTO READYFOREVENT;
        FI;
        CASE HOOKSTATE IS
            WHEN 1 DO
                NEWCALLID = NEWCALLID;
                GOTO CS12BONHOOK;
            WHEN 2 DO
                ABCALLID = NEWCALLID;
                GOTO READYFOREVENT;
            OTHERWISE DO
                EXIT;
            ESAC;
        END;
        EXIT;
    FI;
    IF TADDDATA1 = ZLEGDISCANDTONE THEN
        IF TEVENTORJID = BLEGID THEN
            STATE = CALLSTATE1;
            GOTO READYFOREVENT;
        ELSE
            EXIT;
        FI;
    ELSE
        EXIT;
    FI;
    OTHERWISE DO
        GOTO EXIT;
    ESAC;
EXIT;

```



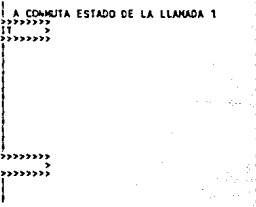
A CONMUTA ESTADO DE LA LLAMADA 12

```

CS12AFASH)
    SPLITRETFLAG = 0;
    SEND SPLIT REFERENCE FSHREF WITH
        FSHREF,
        TASHLAW,
        CS12AFASHREF,
        CLEGID,
        0,
        RESOURCEID;
    EXIT;
CS12AFASH10)
    IF SPLITRETFLAG = 1 GOTO CS12ADONHOOK10;

FREERETFLAG = 0;
SEND FREE REFERENCE FSHREF WITH
    FSHREF,
    TASKCAND;

```



```

CSUSCAMREF,
CLEGID
CS12AF EXIT;
        ASHND;
        IF FREERETFLAG = 1 GOTO CS12AONHOOK20;
        IF FREERETFLAG = 2 GOTO CS12BONHOOK20;
        NLEGID = ALEGID;
        GOTO JOINRELEASE;
    
```

A CUELGA ESTADO DE LA LLAMADA 12

```

CS12AONHOOK)
        SPLITRETFLAG = 1;
        SEND SPLIT REFERENCE FSHREF WITH
            FSHDR
            TASKCAMP
            CSUSCAMREF,
            CLEGID
            RESOURCEID;
        EXIT;
CS12AONHOOK10)
        FREERETFLAG = 1;
        SEND FREE REFERENCE FSHREF WITH
            FSHDR
            TASKCAMP
            CSUSCAMREF,
            CLEGID
            CLEGID = ZNULL;
        EXIT;
CS12AONHOOK20)
        SEND INRETFLAG = 0;
        CLEGID = ZNULL;
        EXIT;
CS12AONHOOK30)
        SEND INRETFLAG = 0;
        SEND SEND IN REFERENCE FSHREF WITH
            SMT
            TASKCAMP
            CSUSCAMREF,
            ALEGID
            CALLDATA
            ZELEMENTB;
        EXIT;
CS12AONHOOK30)
        ALEGONMASK = ZCLEARED;
        BLEGONMASK = ZCLEARED;
        ALEGONMASK.OFFHOOK = ZSET;
        STATE = CALLSTATE12;
        GOTO MONITOREVENT;
    
```

A CUELGA ESTADO DE LA LLAMADA 1

```

>>>>>>>>>>
> SPLIT
>>>>>>>>>>
>>>>>>>>>>
>>>>>>>>>>
>>>>>>>>>>
>>>>>>>>>>
    
```

B ON HOOK CALL STATE 12

```

CS12BONHOOK)
        FREERETFLAG = 2;
        SEND FREE REFERENCE FSHREF WITH
            FSHDR
            TASKCAMP
            CSUSCAMREF,
            BLEGID
            BLEGID = ZNULL;
        EXIT;
CS12BONHOOK10)
        ALEGONMASK = ZCLEARED;
        BLEGONMASK = ZCLEARED;
        CLEGONMASK = ZCLEARED;
        ALEGONMASK.ONHOOK = ZSET;
        CLEGONMASK.OFFHOOK = ZSET;
        STATE = CALLSTATE14;
        GOTO MONITOREVENT;
    
```

```

>>>>>>>>>>
>>>>>>>>>>
>>>>>>>>>>
    
```

3.7.13 ESTADO DE LA LLAMADA 13

A ESCUCHA OCUPADO
B ESTA RETENIDO

IMAGEN

6 -BU-
8 -HU-

EEEEEEEE	SSSSSS	TTTTTTTT	AAAAAA	DDDDDD	OOOOOO		33333
EEEEEEEE	SSSSSS	TTTTTTTT	AAAAAAA	DDDDDD	OOOOOO		3333333
EEEEEE	SSSS	TT	AAAAAAA	DD	OO		33
EEEEEE	SSS	TT	AAAAAAA	DD	OO		333
EEEEEE	SS	TT	AAAAAAA	DD	OO		333

DE	SSS	SSS	TT	AA	AA	DD	DD	DD	DD	DD	DD	DD	DD
DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD
DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD

```

CSTATE:
STATE = CALLSTATE13;
SUBACTION = NONE ;

CASE TEVENT IS

WHEN ZCONMUTA DO
IF TEVENTOBJID = ALEGID
GOTO CS13AF1ASH ;

WHEN ZCONHOOK DO
IF TEVENTOBJID = ALEGID
GOTO CS13ADNHOOK ;
IF TEVENTOBJID = BLEGID
GOTO CS13CONHOOK ;

WHEN ZLEGDISC DO
IF TEVENTOBJID = ALEGID
GOTO CS13ADISC ;
IF TEVENTOBJID = BLEGID THEN
ALEGCONMASK = ZCLEARED ;
BLEGCONMASK = ZCLEARED ;
ALEGCONMASK = ZCLEARED ;
BLEGCONMASK = ZCLEARED ;
ALEGCONMASK.ONHOOK = ZSET ;

STATE = CALLSTATE15 ;
GOTO MONITREVENT ;
FI ;

WHEN ZCALLIMACEALTERADO
IF TADDDATA1 = ZREPLCELEQ THEN
REPLALED = TEVENTOBJID ;
HOOKSTATE1 = TADDDATA2 ;
NEWCALLID = TADDDATA3 ;

IF REPLALED = BLEGID THEN
IF HOOKSTATE1 = A THEN
BLEGID = TADDDATA2 ;
GOTO READYFORCALLID ;
ELSE
HOOKSTATE1 IS
WHEN 1 DO
ABCALLID = NEWCALLID ;
GOTO CS13CONHOOK ;
WHEN 2 DO
ABCALLID = NEWCALLID ;
GOTO READYFOREVENT ;
OTHERWISE DO
ESAC ;
ELSE
EXIT ;
FI ;
ELSE IF TADDDATA1 = ZLEGDISCANDTONE THEN
IF TEVENTOBJID = BLEGID THEN
STATE = CALLSTATE13 ;
GOTO READYFOREVENT ;
ELSE
EXIT ;
FI ;
ELSE
EXIT ;
FI ;
OTHERWISE DO
GOTO EXIT ;
ESAC ;

```

```

-----
ALGUIEN CONMUTA
A CONMUTA
-----
ALGUIEN CUELGA
-----
A CUELGA
B CUELGA
ALGUIEN DESCONECTADO
-----
A DESCONECTADO
-----
IMAGEN ALTERADA
-----

```

```

A CONMUTA ESTADO 13
-----
CS13AF1ASH)
|>>>>>>>>>>>>
|>RELRESOURCES
|>>>>>>>>>>>>
RELRESOURCELAG = 9 ;
SEND RELRESOURCES REFERENCE FSNREF WITH 1 ;
SIMP ;
TASKCAM ;
RESOURCEID ;

EXIT ;
CS13AF1ASH)
IF RESULTCODE /= ZSUCCESS THEN
RESULT = ZFAIL ;
FI ;
RESOURCELAG = ZNULL ;
XLEGID = ALEGID ;
GOTO JOINRELEASE ;

```

```

A CUELGA ESTADO 13
-----
CS13ADNHOOK)
|>>>>>>>>>>>>

```

```

-----
|>>>>>>>>>>>>
|>RELRESOURCES
|>>>>>>>>>>>>
-----

```


3.8.1 CONECTA DOS USUARIOS, LIBERA LA LLAMADA AL SISTEMA Y TERMINA EL USO DE LA FACILIDAD

```

JOINRELEASE)
STATE = JOINRELEASELEG;
IF RESOURCEFLAG = TRUE THEN
  TESTAG = ZTRUE;
ELSE
  TESTAG = ZFALSE;
FI;

SEND JOIN REFERENCE FSHREF WITH
  FSHPTR
  TASECAMP
  CSUSCAMPREF,
  VFCALL,
  ZPLUS,
  ZD1,
  CLECID2,
  TESTAG,
  RESOURCEID;

EXIT;
JOINRELEASE10)
IF RESULTCODE (/= ZSUCCESS THEN
  IF ALEGID2 = BLEGID THEN
    BLEGIDFREE = ZTRUE;
  ELSE
    BLEGIDFREE = ZFALSE;
  GOTO DISCONNECTION;
ELSE
  BLEGIDFREE = ZTRUE;
  CLEGIDFREE = ZTRUE;
  RESULT = ZFAIL;
  GOTO DISCONNECTION;
FI;

CALLID = ICALLID;
STATE = JOINRELEASELEG;

SEND RELCONTROL REFERENCE FSHREF WITH
  FSHPTR
  TASECAMP
  CSUSCAMPREF,
  CALLID;

EXIT;
JOINRELEASE20)
IF RESULTCODE (/= ZSUCCESS THEN
  IF BLEGID2 = BLEGID THEN
    STATE = RELEASECONTROL;
  ELSE
    RESULT = ZFAIL;
    GOTO READYFORVENT;
  BLEGID = BLEGID;
  STATE = RELEASECONTROL;
  RESULT = ZFAIL;
  GOTO READYFORVENT;
FI;

GOTO FINISH;
  
```

3.8.2 FINALIZA EL USO DE LA FACILIDAD DE USUARIO

```

RELRESOURCE)
RELRESOURCEFLAG = 0;
SEND RELRESOURCES REFERENCE FSHREF WITH
  FSHPTR
  TASECAMP
  CSUSCAMPREF,
  RESOURCEID;

EXIT;
RELRESOURCE10)
IF RESULTCODE (/= ZSUCCESS THEN
  RESULT = ZFAIL;
  RESOURCEFLAG = ZNULL;
FI;

FINISH)
RELRESOURCEFLAG = ZTRUE;
IF RESOURCEFLAG (/= ZTRUE GOTO FINISH10;
IF RESOURCEFLAG = ZTRUE GOTO RELDEVICE;
IF BLEGIDFREE = ZTRUE GOTO DISCONNECTION;
IF CLEGIDFREE = ZTRUE GOTO DISCONNECTION;
MULTIPL = ZNULL;
STATE = FINISH;

SEND SENDING REFERENCE FSHREF WITH
  FSHPTR
  TASECAMP
  CSUSCAMPREF,
  CLEGID,
  MULTIPLE,
  ZRESHREE;
  
```



```

=====
3.8.3 ENTREGA EL CONTROL DE LA LLAMADA AL SUBSISTEMA TMS
=====
RELEASE CONTROL 10;
STATE = RELEASECONTROL;
IF AEVENT = ALEGOTSC THEN
    ALEGOTSC = TRUE;
    ALEGID = ALEGID;
    GOTO DISCONNECTION10;
    ALEGOTFREE = ZTRUE;
    GOTO DISCONNECTION10;
ELSE;
    GOTO READYFORAVENT;
FI;
EXIT;
=====
3.8.3 DESCONEXION DE SUBSCRIPTORES
=====
DISCONNECTION10)
STATE = DISCONNECTIONLEGS;
IF ALEGOTFREE /= ZTRUE GOTO DISCONNECTION10;

ALEGOTFREE = ZCLEARED;
IF ALEGID = ZNULL GOTO DISCONNECTION10;

SEND FREE REFERENCE FSHREF WITH
    FSHREF,
    TASKCAUP,
    CSUSCAUREF,
    ALEGID;

ALEGID = ZNULL;
EXIT;

DISCONNECTION10)
IF BLEGOTFREE /= ZTRUE GOTO DISCONNECTION20;

BLEGOTFREE = ZCLEARED;
IF BLEGID = ZNULL GOTO DISCONNECTION20;

SEND FREE REFERENCE FSHREF WITH
    FSHREF,
    TASKCAUP,
    CSUSCAUREF,
    BLEGID;

BLEGID = ZNULL;
EXIT;

DISCONNECTION20)
IF CLEGOTFREE /= ZTRUE GOTO DISCONNECTION30;

CLEGOTFREE = ZCLEARED;
IF CLEGID = ZNULL GOTO DISCONNECTION30;

SEND FREE REFERENCE FSHREF WITH
    FSHREF,
    TASKCAUP,
    CSUSCAUREF,
    CLEGID;

CLEGID = ZNULL;
EXIT;

DISCONNECTION30)
IF NOT DLEGOTFREE = ZTRUE THEN
    NOTHINGIDFREE = ZCLEARED;
FI;

GOTO FINISH;
=====
3.8.4 MONITOREA LAS ACCIONES DEL SUBSCRIPTOR
=====
MONITOREVENT)
IF CLEGID = ZNULL GOTO MONITOREVENT10;
IF CLEGONMASK = ZCLEARED GOTO MONITOREVENT10; 1

NUMBEROFLEGS = 1;
SEND MONITOREVENT REFERENCE FSHREF WITH 1
    FSHREF,
    TASKCAUP,
    CSUSCAUREF,
    NUMBEROFLEGS,
    CLEGONMASK,
    CLEGID;
=====

```



```

        ENABLE INTERRUPT;
    ELSE
        STATE = IDLE;
    FI;
END IDLELISTINSERT;
=====
3.9.3 CONSTRUCCION SEPARADA DE UNA LISTA DE LIBRES
=====
ENTRADA: TASKCAMP, CFIRST
SALIDA: CFIRST, CLAST

ESTA RUTINA CONSTRUYE UNA LISTA DE LIBRES SEPARADA QUE SERA USADA DURANTE EL
INCREMENTO DEL ARCHIVO DE USUARIO, ESTA LISTA ES POR LO TANTO ENLAZADA A
LA LISTA DE LIBRES ORIGINAL.
=====
BEGIN SEPARATEIDLELIST;
    IF CFIRST = ZMOREIDLE THEN
        CFIRST = TASKCAMP;
        CLAST = TASKCAMP;
    ELSE
        TIEMPTR = TASKCAMP;
        TASKCAMP = CLAST;
        NEXTIDLE = TIEMPTR;
        CLAST = TIEMPTR;
        TASKCAMP = TIEMPTR;
    FI;
    STATE = IDLE;
    NEXTIDLE = ZMOREIDLE;
END SEPARATEIDLELIST;
=====
3.9.4 RECONSTRUCCION DE LA LISTA DE LIBRES
=====
ENTRADA: TASKCAMP, CFIRSTIDLE
SALIDA: CFIRSTIDLE, CLASTIDLE

ESTA RUTINA COLOCA EN LA LISTA DE LIBRES A LOS INDIVIDUOS QUE FUERON LIBERADOS
DURANTE EL INCREMENTO DEL ARCHIVO DE USUARIO.
=====
BEGIN IDLELISTREBUILD;
    DISABLE INTERRUPT;
    IF CFIRSTIDLE = ZMOREIDLE THEN
        CFIRSTIDLE = TASKCAMP;
        CLASTIDLE = TASKCAMP;
    ELSE
        TIEMPTR = TASKCAMP;
        TASKCAMP = CLASTIDLE;
        NEXTIDLE = TIEMPTR;
        CLASTIDLE = TIEMPTR;
    FI;
    STATE = IDLE;
    NEXTIDLE = ZMOREIDLE;
    ENABLE INTERRUPT;
END IDLELISTREBUILD;
=====
3.9.5 ENLACE DE DOS LISTAS DE LIBRES
=====
ENTRADA : CFIRST, CLAST, CLASTIDLE
SALIDA : CLASTIDLE

ESTA RUTINA ENLAZA A LA LISTA DE LIBRES QUE FUE CREADA DURANTE EL INCREMENTO
DEL ARCHIVO DE USUARIO CON LA LISTA DE LIBRES ORIGINAL.
=====
BEGIN LINKIDLELISTS;
    DISABLE INTERRUPT;
    IF CFIRSTIDLE = ZMOREIDLE THEN
        CFIRSTIDLE = CFIRST;
        CLASTIDLE = CLAST;
    ELSE
        TASKCAMP = CLASTIDLE;
        NEXTIDLE = CFIRST;
        CLASTIDLE = CLAST;
    FI;
    ENABLE INTERRUPT;
END LINKIDLELISTS;
=====
3.9.6 MARCA A UN INDIVIDUO COMO LIBRE
=====
ENTRADA: CINDIVID
SALIDA: CFIRSTIDLE, CLASTIDLE

ESTA RUTINA COLOCA A TODOS LOS INDIVIDUOS DE REGRESO A LA LISTA DE LIBRES
DESPUES DE QUE HA OCURRIDO UNA RECARGA DEL SISTEMA.
=====
BEGIN IDLEMARKTASKINDIVIDUALS;

```

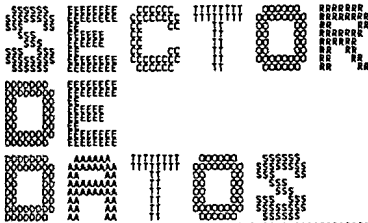
```

ON TASKAMP FROM 0 UPTO CINDUM - 1 DO
DO IDELESTINSEW;
NO;
END IDELEMTASKINDIVIDUALS;
    
```

```

EXIT; EXIT;
END PROGRAM;
DATA;
    
```

4. DATOS INICIALES



4.1 TAMAFI INICIAL DEL ARCHIVO

```

SIZE OF TASKAMP          = 9;
SET FEATUREID = ZCUMBER FOR SUSCOUNTER = 9;
    
```

4.2 DATOS INICIALES

```

SET CCONGCOUNT        = 0;
SET CTRAFICOUNT        = 0;
SET CINDUM              = 0;
    
```

4.3 DIRECCIONAMIENTO DE LAS VARIABLES DENTRO DEL PROCESADOR CENTRAL

```

ALLOCATE ABCALL ID             AT SAC ADDR 01
ALLOCATE ABCALL ID             AT SAC ADDR 02
ALLOCATE ABCALL ID             AT SAC ADDR 03
ALLOCATE ABCALL ID             AT SAC ADDR 04
ALLOCATE ABCALL ID             AT SAC ADDR 05
ALLOCATE ABCALL ID             AT SAC ADDR 06
ALLOCATE ABCALL ID             AT SAC ADDR 07
ALLOCATE ABCALL ID             AT SAC ADDR 08
ALLOCATE ABCALL ID             AT SAC ADDR 09
ALLOCATE ABCALL ID             AT SAC ADDR 10
ALLOCATE ABCALL ID             AT SAC ADDR 11
ALLOCATE ABCALL ID             AT SAC ADDR 12
ALLOCATE ABCALL ID             AT SAC ADDR 13
ALLOCATE ABCALL ID             AT SAC ADDR 14
ALLOCATE ABCALL ID             AT SAC ADDR 15
ALLOCATE ABCALL ID             AT SAC ADDR 16
ALLOCATE ABCALL ID             AT SAC ADDR 17
ALLOCATE ABCALL ID             AT SAC ADDR 18
ALLOCATE ABCALL ID             AT SAC ADDR 19
ALLOCATE ABCALL ID             AT SAC ADDR 20
ALLOCATE ABCALL ID             AT SAC ADDR 21
ALLOCATE ABCALL ID             AT SAC ADDR 22
ALLOCATE ABCALL ID             AT SAC ADDR 23
ALLOCATE ABCALL ID             AT SAC ADDR 24
ALLOCATE ABCALL ID             AT SAC ADDR 25
ALLOCATE ABCALL ID             AT SAC ADDR 26
ALLOCATE ABCALL ID             AT SAC ADDR 27
ALLOCATE ABCALL ID             AT SAC ADDR 28
ALLOCATE ABCALL ID             AT SAC ADDR 29
ALLOCATE ABCALL ID             AT SAC ADDR 30
ALLOCATE ABCALL ID             AT SAC ADDR 31
ALLOCATE ABCALL ID             AT SAC ADDR 32
ALLOCATE ABCALL ID             AT SAC ADDR 33
ALLOCATE ABCALL ID             AT SAC ADDR 34
ALLOCATE ABCALL ID             AT SAC ADDR 35
ALLOCATE ABCALL ID             AT SAC ADDR 36
ALLOCATE ABCALL ID             AT SAC ADDR 37
ALLOCATE ABCALL ID             AT SAC ADDR 38
ALLOCATE ABCALL ID             AT SAC ADDR 39
ALLOCATE ABCALL ID             AT SAC ADDR 40
ALLOCATE ABCALL ID             AT SAC ADDR 41
ALLOCATE ABCALL ID             AT SAC ADDR 42
ALLOCATE ABCALL ID             AT SAC ADDR 43
ALLOCATE ABCALL ID             AT SAC ADDR 44
ALLOCATE ABCALL ID             AT SAC ADDR 45
ALLOCATE ABCALL ID             AT SAC ADDR 46
ALLOCATE ABCALL ID             AT SAC ADDR 47
ALLOCATE ABCALL ID             AT SAC ADDR 48
ALLOCATE ABCALL ID             AT SAC ADDR 49
ALLOCATE ABCALL ID             AT SAC ADDR 50
ALLOCATE ABCALL ID             AT SAC ADDR 51
ALLOCATE ABCALL ID             AT SAC ADDR 52
ALLOCATE ABCALL ID             AT SAC ADDR 53
ALLOCATE ABCALL ID             AT SAC ADDR 54
ALLOCATE ABCALL ID             AT SAC ADDR 55
ALLOCATE ABCALL ID             AT SAC ADDR 56
ALLOCATE ABCALL ID             AT SAC ADDR 57
ALLOCATE ABCALL ID             AT SAC ADDR 58
ALLOCATE ABCALL ID             AT SAC ADDR 59
ALLOCATE ABCALL ID             AT SAC ADDR 60
ALLOCATE ABCALL ID             AT SAC ADDR 61
ALLOCATE ABCALL ID             AT SAC ADDR 62
ALLOCATE ABCALL ID             AT SAC ADDR 63
ALLOCATE ABCALL ID             AT SAC ADDR 64
ALLOCATE ABCALL ID             AT SAC ADDR 65
ALLOCATE ABCALL ID             AT SAC ADDR 66
ALLOCATE ABCALL ID             AT SAC ADDR 67
ALLOCATE ABCALL ID             AT SAC ADDR 68
ALLOCATE ABCALL ID             AT SAC ADDR 69
ALLOCATE ABCALL ID             AT SAC ADDR 70
ALLOCATE ABCALL ID             AT SAC ADDR 71
ALLOCATE ABCALL ID             AT SAC ADDR 72
ALLOCATE ABCALL ID             AT SAC ADDR 73
ALLOCATE ABCALL ID             AT SAC ADDR 74
ALLOCATE ABCALL ID             AT SAC ADDR 75
ALLOCATE ABCALL ID             AT SAC ADDR 76
ALLOCATE ABCALL ID             AT SAC ADDR 77
ALLOCATE ABCALL ID             AT SAC ADDR 78
ALLOCATE ABCALL ID             AT SAC ADDR 79
ALLOCATE ABCALL ID             AT SAC ADDR 80
ALLOCATE ABCALL ID             AT SAC ADDR 81
ALLOCATE ABCALL ID             AT SAC ADDR 82
ALLOCATE ABCALL ID             AT SAC ADDR 83
ALLOCATE ABCALL ID             AT SAC ADDR 84
ALLOCATE ABCALL ID             AT SAC ADDR 85
ALLOCATE ABCALL ID             AT SAC ADDR 86
ALLOCATE ABCALL ID             AT SAC ADDR 87
ALLOCATE ABCALL ID             AT SAC ADDR 88
ALLOCATE ABCALL ID             AT SAC ADDR 89
ALLOCATE ABCALL ID             AT SAC ADDR 90
ALLOCATE ABCALL ID             AT SAC ADDR 91
ALLOCATE ABCALL ID             AT SAC ADDR 92
ALLOCATE ABCALL ID             AT SAC ADDR 93
ALLOCATE ABCALL ID             AT SAC ADDR 94
ALLOCATE ABCALL ID             AT SAC ADDR 95
ALLOCATE ABCALL ID             AT SAC ADDR 96
ALLOCATE ABCALL ID             AT SAC ADDR 97
ALLOCATE ABCALL ID             AT SAC ADDR 98
ALLOCATE ABCALL ID             AT SAC ADDR 99
    
```

