

004.75014

Mar

T

TD 0026-04-01

TD-0026-04-01

Universidad de Oriente



# *Trabajo de Diploma*

**TAC-24K**

Terminal de comunicación para el conmutador digital Infinity TX-2400L/XL, la actualización automática de datos y configuración de servicios telefónicos.

*Autor:*

Adrian Maranje Agramonte

*Tutores:*

Ing. Yadenis Piñeiro Perez.

Ing. José Carlos Miguel Barrios.

Ing. Raydel Montesino Perurena.

Facultad de Matemática y Computación

Ciencia de la Computación

Ciudad de La Habana

Curso 2003-2004

## Resumen

---

Debido al amplio proceso de informatización que se lleva a cabo en la Universidad de las Ciencias Informáticas (UCI) y a la ventaja que en ella se ofrece en cuanto al uso de los medios computarizados, se hace cada vez más necesaria la utilización de estos para la automatización de múltiples actividades.

El presente trabajo constituye una propuesta de un prototipo de un terminal de comunicación para el conmutador digital Infinity TX-2400L/XL de la UCI. Su objetivo esencial es la actualización automática de una base de datos externa (en un ordenador) que servirá para brindar información relacionada con los teléfonos del conmutador a otras aplicaciones o usuarios.

Teniendo en cuenta la información exigida en los reportes de estados por los abonados del conmutador digital Infinity de la UCI, se presentan los siguientes objetivos:

1. Creación de una terminal de comunicación para el conmutador digital Infinity TX-2400L/XL.
2. Creación de una base de datos externa a la central telefónica que será actualizada automáticamente por el terminal anteriormente mencionado.
3. Brindar la posibilidad de que otras aplicaciones o usuarios puedan encuestar dicha base de datos.
4. Habilitar o cancelar algunos servicios telefónicos proveídos por el conmutador Infinity.

La puesta en marcha de esta aplicación proveerá a los usuarios de las líneas telefónicas de una vía rápida para acceder a información relacionada con sus teléfonos. Además de esto esta aplicación será de mucha utilidad para ellos pues constituirá una vía cómoda para configurar algunos servicios contemplados por el conmutador digital Infinity TX-2400L/XL de la UCI desde cualquier rincón de la misma.

## Índice

Declaración de autoría.....	I
Dedicatoria .....	II
Agradecimientos. ....	III
Resumen.....	IV
Índice.....	V
Introducción. ....	1
Capítulo 1 Fundamentación Teórica. ....	4
Introducción .....	4
1.1 El conmutador digital Infinity TX-2400L/XL (central telefónica).....	4
1.1.1 Equipamiento básico .....	5
1.1.2 Unidades principales del conmutador Infinity TX-2400L/XL.....	6
1.2 Los Terminales de Servicio.....	8
1.2.1 Puertos de servicio .....	9
1.2.2 Tipos de Terminales.....	10
1.2.3 Las conexiones.....	10
1.2.4 Algunos terminales de servicios .....	11
1.2.4.1 TXLINK.....	11
1.2.4.2 Hyperterminal de Windows .....	11
1.2.5 Los mensajes de comunicación.....	11
1.3 Herramientas propuestas para el desarrollo del sistema. ....	12
1.3.1 La tecnología .NET. ....	12
1.3.2 Lenguaje de programación.....	13
1.3.2.1 Lenguaje C#. ....	13
1.3.3 Sistemas Gestores de Base de Datos (SGDB).....	14
1.3.3.1 Microsoft SQL Server 2000.....	14
1.3.4 Lenguaje de modelación. ....	15
1.3.4.1 UML (Unified Modeling Language).....	15
Capítulo 2 Características del sistema.....	16
Introducción .....	16
2.1 Objeto de Estudio .....	16
2.1.1 Problema .....	16
2.1.2 Ubicación. ....	18
2.2 Propuestas de sistema.....	19
2.3 Requerimientos del sistema. ....	20
2.3.1 Dependencias y relaciones.....	20
2.3.2 Requisitos funcionales. ....	21
2.3.3 Requisitos no funcionales. ....	22
2.3.3.1 Requisitos de Interfaz externa.....	22
2.3.3.2 Requisitos de funcionalidad.....	22
2.3.3.3 Requisitos de usabilidad.....	22
2.3.3.4 Requisitos de Rendimiento. ....	23
2.3.3.5 Requisitos de confiabilidad.....	23
2.3.3.6 Requisitos de diseño e implementación.....	24

2.3.3.7 Requisitos de software y de hardware.....	24
2.3.3.8 Requerimientos legales. ....	25
2.4 Especificación de los casos de uso.....	25
2.4.1 Identificación de los actores.....	25
2.4.2 Listado de los casos de usos del sistema.....	26
2.4.3 Diagrama de casos de uso .....	29
2.4.4 Casos de uso por ciclos. ....	31
2.4.4.1 Casos de uso del núcleo central. ....	31
2.5 Modelo conceptual.....	32
Capítulo 3 Análisis y diseño del sistema. ....	34
Introducción. ....	34
3.1 Extensión de los casos de uso .....	34
3.2 Diagramas de interacción de los casos de uso .....	34
3.3 Diagrama de clases.....	35
3.4 Diseño de la base de datos .....	36
Capítulo 4 El TAC-24K .....	37
Introducción. ....	37
4.1 Descripción general de TAC-24K.....	37
4.2 Comunicación con el conmutador.....	38
4.3 Base de datos del sistema.....	39
4.4 Interfaz gráfica para la ejecución de las tareas.....	40
4.5 Las tareas automáticas.....	41
4.6 Las actividades concurrentes.....	42
4.7 Mensajes recibidos y servicios configurables por TAC-24K. ....	42
4.7.1 Identificación y facturación de llamadas.....	43
4.7.2 Servicio de despertador automático .....	46
4.7.3 Servicio de no molestar.....	47
4.7.4 Mensaje de difusión. ....	48
4.7.4.1 La hora desde el conmutador digital. ....	49
Conclusiones. ....	50
Recomendaciones.....	52
Referencias bibliográficas.....	53
Bibliografía .....	54
Anexos .....	55
Anexo 1. TX-2400L/XL con dos gabinetes. ....	55
Anexo 2. Requerimiento de hardware para el desempeño de la aplicación.....	56
Anexo 3. Extensión de casos de uso implementados.....	57
Anexo 4. Diagramas de clases e interacción.....	65
Anexo 5. Descripción de las clases de diseño.....	84
Anexo 6. Descripción de las tablas de la base de datos. ....	95
Anexo 7. Diseño lógico y físico de la base de datos.....	100
Glosario de términos. ....	101

## Introducción.

---

Es evidente el cambio que ha sufrido en los últimos años el uso de las tecnologías de la telecomunicación. Cada vez se hace más necesario la utilización de vías rápidas y cómodas para atender las grandes demandas de los usuarios que día a día hacen uso de estas tecnologías.

En la actualidad son pocos los países que pueden hacer grandes inversiones para tratar de solucionar el problema anteriormente planteado. Cuba se encuentra en el grupo de estos países y debido a las restricciones económicas y políticas a las que debemos enfrentarnos en estos tiempos nos vemos en la necesidad de desarrollar con nuestros propios esfuerzos y recursos lo que por otra vía nos es casi imposible conseguir.

Grandes son los avances que se han obtenido en el plano de las telecomunicaciones en nuestro país pero estos aún no son suficientes. Por ejemplo, hoy podemos hablar de que en Cuba la mayoría de las centrales telefónicas que operan en nuestro sistema de comunicación están equipadas con altas tecnologías digitales; sin embargo las grandes posibilidades que brinda esta tecnología aún no han sido explotadas debido a la poca experiencia que se tiene con el uso de ellas.

Uno de los pasos que se han dado en los últimos años en cuanto al desarrollo de las tecnologías de las telecomunicaciones y la informática en Cuba ha sido la creación de la Universidad de las Ciencias Informáticas (UCI), la cual lleva a cabo un amplio proceso de automatización e informatización de todo lo referente al uso de las Técnicas de la Informática y las Comunicaciones (TIC). Se espera que en un futuro no muy lejano cuando todo el país pueda contar con los medios que cuenta la UCI, se puedan aplicar a nivel nacional las experiencias que se comenzarán experimentar en dicho centro.

Uno de los aspectos tenidos en cuenta en el equipamiento y prestación de servicios de la UCI lo constituye sin duda alguna la prestación de un servicio de

calidad en el plano de las telecomunicaciones. Para ello se ha implantado en dicho centro una central telefónica digital privada modelo Infinity TX-2400L/XL de fabricación canadiense. Entre las características esenciales de este tipo de central telefónica se encuentran la posibilidad que les brinda a los usuarios de sus líneas telefónicas de poder utilizar múltiples servicios que normalmente proveen otras sucursales de las telecomunicaciones. Entre estos servicios se encuentran el matutino o despertador, el enrutamiento de llamadas, el bloqueo de llamadas; así como también la posibilidad de conocer en que estado se encuentran sus líneas telefónicas, obtener información sobre las llamadas realizadas o recibidas en sus teléfonos, etcétera.

A pesar de las potencialidades del conmutador Infinity TX-2400L/XL y del equipamiento computarizado del que dispone la UCI, aún no se cuenta con una vía rápida, sencilla y segura que permita a sus abonados realizar las operaciones anteriormente mencionadas. Actualmente la única vía posible consiste en la solicitud directa al administrador de la central telefónica de la UCI de reportes concernientes a las llamadas entrantes y salientes de una línea en específico o de las facturación de las mismas, etcétera. A esto se suma que la mayoría de los abonados en ocasiones necesitan configurar determinados servicios a sus teléfonos, operación casi siempre imposible de realizar si no se hace desde el mismo teléfono o mediante la intervención del administrador bajo previa solicitud del abonado de forma verbal.

Teniendo en cuenta las limitantes y dificultades anteriormente mencionadas se plantea como objetivo general de este proyecto:

- Automatizar el proceso de presentación de reportes de información relacionada con las líneas telefónicas de la central telefónica Infinity TX-2400L/XL instalada en la UCI.
- Brindar la posibilidad de configurar servicios telefónicos desde cualquier punto de la universidad.

A partir del análisis del objetivo general se derivan los siguientes objetivos específicos:

- Confección de una aplicación o software que constituirá un terminal de comunicación para la conmutador digital Infinity TX-2400L/XL o central telefónica que será el encargado de almacenar la información relacionada con las líneas telefónicas que provee dicha central utilizando los mensajes de notificación de sucesos del conmutador.
- Creación de una base de datos externa a la central telefónica de la UCI que contendrá información relativa a las líneas telefónicas del conmutador y que será actualizada de manera automática por el terminal propuesto.
- Brindar un mecanismo de acceso remoto para la configuración de algunos servicios telefónicos a los abonados de la central telefónica de la UCI.

El presente documento se estructura en cuatro capítulos y anexos, que incluyen todo lo relacionado con el trabajo investigativo realizado, así como también el análisis y el diseño de la herramienta propuesta para la realización del mismo.

El Capítulo 1 *Fundamentación Teórica* recoge el análisis del tema a tratar y de las tecnologías actuales relacionadas con el tema. Se incluye además una breve descripción de las herramientas propuestas para la implementación y modelación.

El Capítulo II *Características del sistema* describe el objeto de estudio, el entorno de trabajo en que se desarrolla el sistema, la propuesta del sistema, requerimientos funcionales y no funcionales y los casos de uso del sistema.

El Capítulo III *Diseño del Sistema* muestra la expansión de los casos de uso, el modelo conceptual y el modelo diseño. Se plantean los detalles relacionados con el diseño de la base de datos que se necesita.

El Capítulo IV *El TAC-24K* describe el proceso de operación del terminal de comunicación propuesto, el uso de la concurrencia para evitar la pérdida de datos y cada uno de los mensajes de comunicación empleados para el intercambio de información con el conmutador Infinity TX-2400L/XL.

# Capítulo 1

## Fundamentación Teórica.

---

### Introducción

El presente capítulo está destinado a brindar una breve descripción de algunos aspectos relacionados con el conmutador digital o central telefónica Infinity TX-2400L/XL, enfatizando en las características y procesos que están estrechamente vinculados con el desarrollo del trabajo. Se ofrecerá además una descripción de los mecanismos de comunicación empleados en el trabajo y como un tercer aspecto se hará un breve estudio acerca del lenguaje de programación propuesto para el desarrollo de este proyecto.

### 1.1 El conmutador digital Infinity TX-2400L/XL (central telefónica)

**TX-2400L/XL** es un conmutador muy moderno y versátil para aplicaciones Privadas, Rurales y Públicas. El sistema básico es un conmutador de cambio digital que provee conexiones de modulación de códigos de pulsos (MCP) para los caminos de voz y de datos entre puertos de tipos diversos todo esto bajo el control de un programa de centralización almacenado.

**TX-2400L/XL** puede proveer a hasta 3,840 puertos Digitales/Analógicos sin bloqueo ni concentración. El conmutador de MCP de red de TX-2400L/XL tiene una configuración distribuida. Cada estante tiene su propio conmutador residente de MCP y sus líneas pasivas, todo bajo el control de la Unidad de Control Central (CCU) que conecta todas las líneas de los estantes [TX-Communication Canadá Inc].

En el anexo 1 se muestra una imagen del conmutador digital Infinity TX-2400L/XL como el que se encuentra brindando servicio en la UCI.

### **1.1.1 Equipamiento básico**

El sistema infinity se construye en cuatro formularios diferentes a modo de armarios o gabinetes en pie para acomodar las distintas componentes del mismo.

El TX-2400L/XL es el más grande los sistemas de Infinity, consiste en un gabinete principal y dos gabinetes de expansión. Cada gabinete acoge cuatro líneas de estantes. Cada línea de estantes tiene una capacidad para 320 puertos.

El TX-2400L es similar al TX-2400L/XL pero a diferencia de este consta de un gabinete principal y solamente uno de expansión, cada gabinete del TX-2400L acomoda cuatro líneas de estantes y cada línea acomoda 320 puertos.

El TX-200M es la versión media de Infinity. Este sistema consta de un gabinete principal y uno de expansión. Cada gabinete acomoda 320 puertos.

El TX-2400S es la versión más pequeña de la familia Infinity. Consta de un gabinete principal y uno de expansión. Cada gabinete acomoda 96 puertos.

Cada línea de estante puede acomodar hasta 320 puertos los cuales pueden ser líneas telefónicas o troncos en dependencia del módulo del circuito utilizado. Un máximo de 1024 conversiones triples pueden existir en el sistema en un momento dado incluyendo 16 conferencias por cada estante.

En el caso del TX-2400L/XL y el TX-2400L, el gabinete principal acoge el estante de Control Común y hasta cuatro líneas de estantes más, mientras que en el TX-2400S el gabinete principal contiene un simple estante que funciona como estante de control y estante de líneas a la vez. Toda la energía de los gabinetes, los datos y los caminos de discursos son transportados por cables.

Todos los cables de las conexiones se proporcionan por detrás de los gabinetes excepto la alimentación y la tierra.

Los gabinetes son refrescados por cuatro ventiladores en la parte superior trasera de los mismos. El aire es extraído desde el frente de los gabinetes hacia la parte trasera a través de las aberturas de la parte superior trasera [1].

### **1.1.2 Unidades principales del conmutador Infinity TX-2400L/XL**

El estante de control contiene circuitos que proveen el control común y el estado de espera optativo del funcionamiento del sistema. Estos incluyen la generación y distribución de señales de tiempo, datos de Entrada/Salida a las líneas de los estantes, decisiones tomadas y la interacción con el equipo y el personal de servicio.

Los estantes de líneas son unidades modulares que proporcionan acceso a los grupos de puertos de voces y subsistemas de control. Cada estante contiene un equipamiento local de control común, líneas, troncos e interfaces de control. Los conectores son proveídos por la parte trasera de los estantes de líneas por cables de 25-par por el marco de distribución.

La unidad principal del sistema Infinity TX-2400L/XL es la Unidad de Control de Servicio (SCU) provee muchas funciones importantes y necesarias para la operación del sistema, el mantenimiento y el control. Es un módulo individual que reside en el estante de control del sistema y en el estante común de control de línea del sistema Infinity. Las principales funciones de la SCU son las siguientes:

1. La SCU provee la interfaz del sistema hombre-máquina a través de dispositivos periféricos de terminal y módem, permitiendo la interacción del equipo de servicio con el sistema del procesador, para monitorear o cambiar la base de datos del programa de control y dar información sobre la operación del sistema (las estadísticas de tráfico, los defectos, etc.).

2. Las condiciones de defecto generadas por las diversas tarjetas del circuito del sistema son aplicadas directamente a la SCU la cual consecuentemente genera las señales de alarma.

El programa de control de la SCU es un programa almacenado en una tarjeta de memoria ROM (Read Only Memory) o en una tarjeta de memoria EPROM/OTP. El programa de control, en conjunción con los diversos dispositivos periféricos realiza las siguientes funciones:

1. La SCU provee el cargador de arranque en frío del sistema (BOOT) a través de una unidad de discos floppy o de la interfaz de la tarjeta de memoria. La unidad de discos floppy es accesible a través de la Interfaz de Periféricos (SPI) de la SCU, la cuál consta de un bus genérico del multiplexor que conecta la ranura de la tarjeta de la unidad de disco floppy con la SCU. La interfaz de la tarjeta de memoria puede servir para la carga y configuración del sistema o como RAM añadida al programa.
2. Proveer a los empleados de servicio de una manera para controlar y modificar la operación del sistema y para mostrar el estado de operación del sistema.
3. Recibir las condiciones de fallos de diversos circuitos del sistema y generar señales de alarma consecuentemente.
4. Comunicación con las Unidades de Centrales de Control (CCUs) a través buses del sistema.

La SCU interactúa con la unidad de disco floppy por medio de un bus genérico del multiplexor posibilitando la entrada de cualquier tarjeta en la ranura de la unidad de disco floppy para comunicarse con la SCU. Principalmente la unidad de disco floppy será utilizada como una ranura periférica de la tarjeta conteniendo unidades de floppys, discos duros, drivers, puertos series y puertos paralelos. La información de configuración puede ser tele-enviada al sistema a través de esta

interfaz. La información de contabilidad de llamada puede ser almacenada en dispositivos de memoria no volátil para poder recuperarla en caso necesario.

El conmutador digital Infinity TX-2400L/XL cuenta con las tarjetas de Interfaz Universal de Línea (ULI) e Interfaz Universal Digital (UDI) que son empleadas para acomodar las distintas aplicaciones analógicas o digitales.

La tarjeta ULI consta de circuitos para:

- Líneas telefónicas estándar
- Teléfonos de pago (Cajas recaudadoras)
- Teléfonos de magneto
- Marcado directo interno

La tarjeta UDI, por su parte, con el equipamiento adecuado provee circuitos para:

- Operador de consola.
- Teléfonos.
- Troncos.

## 1.2 Los Terminales de Servicio

El sistema Infinity utiliza programas genéricos idénticos proporcionados en tarjetas EPROMs no volátiles o en el caso del TX-2400S en Memoria de Flash. Cada instalación puede programarse con su propia base de datos única generada para satisfacer los requerimientos individuales. La base de datos reside en la memoria de acceso de aleatorio, permitiendo su fácil modificación por parte del usuario. También existen medios para agregar módulos de memoria en caso que se desee expandir la misma.

Las funciones para la administración (cambio de número de directorio, Clases de Servicios, enrutamiento de tronco, etcétera) puede ser introducidas a través de un terminal de servicio. El terminal de servicio puede estar conectado de forma local o

remota a través de un modem. La administración puede llevarse a cabo a través de un operador de consola.

Cuando los fallos mayores son detectados son activadas las alarmas en las consolas y en los circuitos externos. El personal de servicio debe guiarse por las indicaciones visibles en el equipo defectuoso o por la interacción con las funciones de mantenimiento de la Unidad de Control de Servicio (SCU). La información de fallo es almacenada y puede ser impresa en cualquier momento. El sistema puede ser programado para reportar automáticamente los fallos a una localización central [2].

Los Terminales de Servicios son utilizados para examinar o modificar la programación del conmutador digital Infinity así como para diagnosticar los problemas en el funcionamiento del sistema. La interfaz de la estación de grabación de detalles de mensajes (SMDR - Station Message Detail Recording) se usa para mostrar información de las llamadas. El terminal de servicio y la SMDR se unen localmente y/o remotamente con el sistema a través de la tarjeta SCU. Localmente esta conexión se realiza a través de una interfaz RS-232C por medio de un conector de pines estándar EIA-25 situado en el panel trasero del gabinete. Dos modem data/fax en la tarjeta SCU proporcionan la interfaz remota [TX/Communication Canadá Inc].

Las órdenes para el sistema son introducidas por el terminal de servicio. Cualquier respuesta resultando de estas órdenes es mostrado en el terminal de servicio. Los terminales de servicio pueden ser unidades sencillas de despliegue o programas ejecutándose en computadoras personales comunicándose a través del puerto serie.

### **1.2.1 Puertos de servicio**

El TX-2400M y el TX-2400L/XL poseen tres conectores de hardware RS-232 ubicados en la parte trasera del estante de control para posibilitar a los equipos de servicios interactuar con los puertos de la Unidad de control de Servicio. Las

conexiones son para el terminal de servicio, los reportadores de cuentas de llamadas (SMDR) y una conexión RS-232C adicional. El TX-2400s posee conectores de hardware RS-244.

Las conexiones con el terminal de servicio y Modem son usadas para introducir comandos para imprimir o cambiar la configuración del sistema, etcétera y pueden tener ubicación remota (conexión por modem) o ubicación local (Terminal de Servicio).

El puerto SMDR es usado para obtener detalles de las llamadas en disco duros, terminales de video o el software TX Call Accounting [3].

## 1.2.2 Tipos de Terminales

Cualquier dispositivo terminal de datos (incluyendo terminales de copia en papel) que cuente con los siguientes requisitos puede ser usado como un terminal de servicio:

- Mostrar/Imprimir el conjunto de caracteres ASCII
- Generación de caracteres en mayúscula (capacidad de CAPS LOCK)
- Interfaz serie RS-232C

## 1.2.3 Las conexiones

El terminal de servicio puede estar localizado en el propio conmutador de TX-2400L/XL o en una localización remota a través del uso de módems. La conexión local del terminal de servicio con el TX-2400L/XL será por uno de los puertos localizados en el panel anterior del gabinete. La conexión remota será a través de uno de los dos puertos vía modem ubicados al dorso del panel. Para comunicar uno con otro la configuración de la comunicación a través del puerto del TX-2400L/XL y del terminal de servicio deben ser idénticas. La configuración predeterminada del procesador de servicio del TX-2400L/XL para la comunicación a través del puerto serie es la siguiente:

- Velocidad de transmisión de 9600 bits por segundo
- 8 bits de datos
- Sin paridad
- 1 bits de parada

## **1.2.4 Algunos terminales de servicios**

### **1.2.4.1 TXLINK**

TEXAS Link es un programa informático propietario que permite a una Computadora personal de IBM (PC) o dispositivo compatible operar como un terminal de servicio. A diferencia de un terminal “mudo”, usar una PC permite el almacenamiento de los datos de configuración de TX-2400L/XL en disco. En el caso de una falla eléctrica de largo plazo, los datos de configuración pueden ser vueltos a cargar desde la PC para el TX-2400L/XL.

### **1.2.4.2 Hyperterminal de Windows**

Se trata de una aplicación de comunicaciones contenida en las versiones más recientes del sistema operativo Windows. Permite, por ejemplo, conectarse por vía telefónica a boletines electrónicos y descargar o cargar programas y documentos.

HyperTerminal sirve también para transferir archivos grandes de un equipo a un equipo portátil a través del puerto serie, en lugar de realizar la configuración del portátil en una red. También puede utilizar HyperTerminal para comunicarse con los equipos antiguos basados en caracteres.

## **1.2.5 Los mensajes de comunicación**

Los mensajes de comunicación no son más que una secuencia de bytes que se envían desde el conmutador digital Infinity TX-2400L/XL a un terminal de servicio, conectado al puerto SMDR del conmutador, y de este último al conmutador digital. Cada mensaje tiene una característica específica en dependencia de la información que contenga.

El formato general de los mensajes es el siguiente:

byte 1	STX (2)	Comienzo del mensaje
byte 2,3		Código de la función
byte 4,5		Código del proceso
byte 6...n		Datos adicionales específicos para cada código de función.
byte n+1		Número de Secuencia del Mensaje (MSN)
byte n+2,3	BCC	Bloque de chequeo de caracteres el cual es un OR exclusivo de todo el texto en bytes excluyendo el byte de comienzo y el byte de fin del mensaje.
byte n+4	ETX (3)	Fin del mensaje

El contenido de los bytes de los mensajes transmitidos o recibidos corresponde a códigos ASCII de caracteres.

## **1.3 Herramientas propuestas para el desarrollo del sistema.**

Para el desarrollo del sistema se realizó un estudio sobre las posibles herramientas a utilizar en su construcción. Teniendo en cuenta la tendencia actual y las novedades en este campo se hace la siguiente propuesta.

### **1.3.1 La tecnología .NET.**

Microsoft .NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir software en forma de servicios, de modo que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.

El Visual Studio .Net posee una extensa librería de clases que proporciona soporte para acceso a datos (relacionales y XML), servicios de directorio, colas de mensajes, etc. Posee librerías de clases bases (Base Class Library) que contiene las clases necesarias para realizar las tareas más comunes como por ejemplo manipulación de cadenas de caracteres, programación multi-hilo, etcétera. Es una plataforma neutral al lenguaje, y por diseño, neutral al sistema operativo. Se puede usar el lenguaje con el que se sea más productivo sin ningún tipo de desventaja con respecto a usar otro de los muchos disponibles para la plataforma.

Con el objetivo de satisfacer las necesidades de las empresas desarrolladoras de software, Microsoft construyó la herramienta para el desarrollo rápido de aplicaciones y servicios Web: Visual Studio .NET. En ella se incluyen lenguajes tales como el Visual Basic, J#, C++ y el nuevo lenguaje C#, y trabaja por incluir una serie de nuevos lenguajes de programación.

### **1.3.2 Lenguaje de programación.**

C y C++ son dos de los lenguajes más utilizados en el campo de la ingeniería y la programación de sistemas. La principal razón es que C y C++ proporcionan el nivel de abstracción preciso para construir una aplicación compleja, pero, al mismo tiempo, ofrecen mecanismos de bajo nivel que permiten a los programadores hacer uso de las características más avanzadas de las plataformas sobre las que se ejecutan sus programas. Microsoft ha creado C# que combina algunas de las características más avanzadas de Java con algunas de las más potentes de C y C++. La idea es convertirlo en el nuevo lenguaje de Internet y, por supuesto, en el lenguaje nativo para acceder a todos los servicios que en el futuro brindará .NET.

#### **1.3.2.1 Lenguaje C#.**

C# es el lenguaje de .NET construido especialmente para adaptarse de manera natural al framework y aprovechar al máximo todas sus características. Al igual que C y C++, permite programar fácilmente a bajo nivel, este incorpora características encontradas en los lenguajes industriales y de investigación más

habituales. Gracias a esto, acceder a las características avanzadas de la plataforma sobre la que se trabaje, crear código muy eficiente en aquellos puntos de la aplicación que son críticos y acceder a las Interfaces de Programación de Aplicaciones (APIs) existentes es perfectamente posible. Muchos dicen que si Java se puede considerar un C++ mejorado en cuestiones de seguridad y portabilidad, C# debe entenderse como un Java mejorado en todos los sentidos.

### **1.3.3 Sistemas Gestores de Base de Datos (SGDB).**

Una Base de Datos (BD) es un conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos relacionados, variables en el tiempo [MATOS 99]. En la actualidad existen numerosos sistemas gestores de bases de datos, entre ellos el Microsoft Access, Oracle, MySQL, Visual Fox Pro, etc., sin embargo se seleccionó el Microsoft SQL Server 2000 por las ventajas que ofrece.

#### **1.3.3.1 Microsoft SQL Server 2000.**

Microsoft SQL Server 2000 es un Servidor de Base de Datos y herramienta de Análisis de la información. Proporciona seguridad, fiabilidad y escalabilidad para poner en marcha cualquier aplicación en un tiempo pequeño, destacando sus sencillas tareas de administración y su capacidad de analizar la información.

SQL Server permite la creación de procedimientos almacenados, los cuales consisten en instrucciones que se almacenan dentro de una base de datos de SQL Server, realizados en lenguaje SQL. Son procedimientos que se guardan semicompilados en el servidor y que pueden ser invocados desde el cliente. Se ejecutan más rápido que instrucciones SQL independientes [SQL].

## **1.3.4 Lenguaje de modelación.**

### **1.3.4.1 UML (Unified Modeling Language)**

El Lenguaje de Modelación Unificado (UML - Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar, construir y documentar cada una de las partes que comprende el desarrollo de software. Posee formas de modelar conceptos como por ejemplo las funciones del sistema, además de otras particularidades como la de escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. Usa procesos de otras metodologías, aprovechando la experiencia de sus creadores [UML 03]. Es desde finales de la década del 90, un lenguaje de modelación orientado a objetos estándar, de acuerdo con el Object Management Group, siendo utilizado diariamente por grandes organizaciones como: Microsoft, Oracle, Rational, etc. Existen herramientas Case de trabajo visuales como el Analise, el Designe, el Rational Rose, que permiten realizar el modelado del desarrollo de los proyectos. En la actualidad una de las mejores y más utilizadas en el mercado mundial que provee el modelado visual basado en UML es Rational Rose por lo que es la usada en la modelación de este proyecto.

# Capítulo 2

## Características del sistema.

---

### Introducción

A través de este capítulo se describe el objeto de estudio, el entorno de trabajo en que se desarrolla la aplicación, se realiza la propuesta del sistema y se analizan los requerimientos funcionales y no funcionales así como los casos de uso del sistema.

### 2.1 Objeto de Estudio

#### 2.1.1 Problema

Con el objetivo de mantener una estrecha comunicación entre sus habitantes y de estos con el resto del país, la Universidad de las Ciencias Informáticas (UCI) se ha dado a la tarea de instalar al menos una línea telefónica en cada uno de los locales donde pueda encontrarse un trabajador o estudiante.

Actualmente la UCI cuenta con una central telefónica modelo Infinity TX-2400L/XL la cual brinda diversos servicios a sus abonados a través de los teléfonos que se instalan en dicho centro. Como es conocido por todos el número de locales construidos en la UCI y que actualmente cuentan con al menos una línea de teléfono es considerablemente grande sin contar los que aun están por construir. Actualmente se encuentran instalados y brindando servicio aproximadamente 1200 líneas telefónicas todas conectadas al conmutador digital Infinity TX-2400L/XL de la UCI. Por tal motivo, cada vez, se hace más necesaria una mayor rapidez en cuanto a la atención de las solicitudes de información provenientes de los abonados por parte del personal encargado de atender dichas peticiones.

El proceso actual establecido en la universidad para la gestión de información referente a las comunicaciones telefónicas, dígame estado del teléfono, llamadas recibidas, etcétera; se hace de forma manual y es realizado por una sola persona que es además la encargada de administrar la referida central telefónica mientras que la configuración de los distintos servicios que el conmutador Infinity puede brindar, entre ellos el despertador, el bloqueo de llamadas, etcétera; solo puede realizarse desde la propia línea telefónica. Las solicitudes de información son tramitadas por el administrador encuestando directamente al conmutador digital a través de una aplicación proveída en el paquete de instalación de Windows, el HyperTerminal utilizando para ello comandos de líneas al estilo de los del Sistema Operativo MS-DOS. Estos comandos, en su mayoría, son un tanto incómodos de utilizar debido a la sintaxis de escritura de los mismos. A medida que crece la universidad en locales y en líneas telefónicas crece también la demanda de los abonados referente al tema de la información telefónica por lo que cada vez se hace más difícil para el administrador de la central cumplir en tiempo con todas las solicitudes que a diario recibe.

A primera vista pudiera pensarse que una solución a este problema pudiera ser la contratación de más personal para la atención de las solicitudes de los abonados y de esta forma concentrar al administrador de la central telefónica en aquellas actividades que solo conciernen al correcto funcionamiento de la misma, sin embargo, esta aparente solución no es ni siquiera viable debido a que la comunicación entre el HyperTerminal y el conmutador digital Infinity TX2400L/XL se realiza a través del puerto serie de una computadora personal y este tipo de comunicación solo permite una aplicación conectada en cada extremo de la conexión. Además a este inconveniente se suman las exigencias que debe cumplir todo aquel que pretenda manipular una central telefónica de este modelo ya que para ello es necesario poseer los conocimientos adecuados poder operar una central telefónica de este tipo.

Una limitante adicional que acompaña al problema anterior la constituye sin duda alguna la imposibilidad por parte de los abonados de poder configurar servicios telefónicos a sus teléfonos desde una localización que no sea desde el propio teléfono.

Los problemas anteriormente planteados traen aparejados la creciente necesidad de una herramienta automática que sea capaz de atender las solicitudes antes mencionadas para permitir al administrador de la central realizar aquellas operaciones que están encaminadas a lograr el funcionamiento óptimo de la misma y para posibilitarle a los abonados que pueden desde cualquier punto de la universidad configurar determinados servicios a sus teléfonos.

### **2.1.2 Ubicación.**

La Universidad de las Ciencias Informáticas (UCI) constituye la universidad más joven del país, creada con el objetivo de formar a personal altamente calificado en ingeniería informática, que contribuya en gran medida al desarrollo de ese campo y de la economía del país, y que además, “soporte a la informatización de la sociedad cubana y la competitividad internacional de la industria cubana del software” [UCI 03].

En estos momentos se está desarrollando en ella el Proyecto UCI-Ciudad Digital, que tiene como misión “informatizar la UCI y crear una cultura digital, utilizando las tecnologías más novedosas del estado del arte para lograr la integración y funcionamiento armónico de todos los procesos, desarrollando así el prototipo para la informatización de la Sociedad Cubana” [MACHADO 03].

Para lograr estos objetivos se ha dividido el proyecto general en áreas de trabajos entre la cuales figura la de telecomunicaciones. Dicha área tiene la misión de automatizar la gestión de la información sobre las líneas telefónicas de la Universidad, así como proveer a la misma de otra vía para la habilitación de algunos de los servicios telefónicos que puede brindar la central telefónica instalada en este centro.

La herramienta propuesta en este trabajo no solo podrá ser utilizada en la UCI sino que además podrá emplearse en cualquier otra entidad que cuente con una central telefónica modelo Infinity TX-2400L/XL. Por tal motivo la aplicación encargada de recoger la información y atender las peticiones de servicio se hará de manera que pueda configurarse y adaptarse a cualquier otra entidad como por ejemplos los hoteles del turismo internacional que cuentan con este tipo de equipamiento.

## 2.2 Propuestas de sistema

Esta aplicación esta diseñada con el objetivo de dotar a la UCI de una vía que permita a los abonados, de manera fácil, rápida y cómoda, acceder a información referente a su(s) línea(s) de teléfono así como también brindarles otra vía para la configuración de servicios telefónicos.

La gestión de la información se realizará a través de una base de datos localizada en un ordenador y la misma será actualizada periódica y automáticamente por el terminal propuesto en este documento.

Para la configuración de los servicios telefónicos por parte de los abonados a través del terminal se utilizará la misma base de datos. En ella se recogerá la información necesaria para la configuración de determinados servicios proveídos por el conmutador digital Infinity TX-2400L/XL. El terminal propuesto se encargará de recoger de forma periódica las solicitudes de los abonados registradas en la base de datos y procederá a su configuración en el conmutador.

Para la comunicación entre el conmutador Infinity y el terminal propuesto se establece una conexión a través de uno de los puertos series (RS-232C) de una computadora con el puerto SMDR ubicado en la parte posterior del panel de comunicación del conmutador Infinity. Los parámetros de esta comunicación deben ser idénticos para los dos terminales presentes en la comunicación con el objetivo de evitar pérdidas o envío y recepción errónea de datos.

Atendiendo a las características de la comunicación perpetua que hay que existir para el intercambio de información entre la computadora y el conmutador digital Infinity, el terminal propuesto se dividió en dos módulos. Las funcionalidades de cada uno de ellos son como sigue:

- Módulo de lectura y actualización de datos: es el encargado de recoger la información proveniente del conmutador Infinity y almacenarlos en la base de datos concebida en una computadora.
- Módulo de configuración de servicios: es el encargado de recoger la información almacenada en la base de datos, la cual contiene los datos necesarios para la habilitación o cancelación de algunos servicios telefónicos solicitados por los abonados.

## **2.3 Requerimientos del sistema.**

Los requerimientos del sistema constituyen las condiciones indispensables que debe poseer cualquier sistema para su correcto funcionamiento.

### **2.3.1 Dependencias y relaciones**

Para lograr una óptima ejecución del software prototipo propuesto en este trabajo es necesario la presencia de una serie de factores importantes de los cuales se vale dicho software para realizar las operaciones para la cual fue concebido. Entre ellos se encuentran:

- Aplicación Servidor o cliente de SQL para Windows para el acceso a la base de datos.
- Framework de .Net para la ejecución del Terminal de Servicio en la computadora.
- Cable de conexión con un conector para puerto EIA-9 en el extremo de la computadora y uno EIA-25 en el extremo del conmutador.

La siguiente tabla muestra las conexiones de los pin en el cable mencionado anteriormente:

PC		Conmutador Infinity	
9-pin		25-pin	
3	TXD ----->-----	RXD	2
2	RXD -----<-----	TXD	3
7	RTS ----->-----	CTS	4
8	CTS -----<-----	RTS	5
6	DSR -----<-----	DTR	6
5	SG -----	SG	7
1	DCD -----<-----	DCD	8
4	DTR ----->-----	DSR	20

### 2.3.2 Requisitos funcionales.

Para el cumplimiento de los objetivos específicos del proyecto se plantean los siguientes requerimientos funcionales:

1. Conexión con la planta telefónica Infinity
  - 1.1 Establecer parámetros de la conexión utilizando el protocolo de comunicación TCP/IP (servidor y puerto)
  - 1.2 Establecer los parámetros de conexión con la base de datos especificando nombre del servidor de datos y nombre de la base de datos.
2. Diseñar e implementar una base de datos para almacenar toda la información obtenida.
  - 2.1 Dar mantenimiento a la base de datos.
  - 2.2 Actualizar los datos de los teléfonos instalados así como el estado de los servicios de los mismos.
3. Obtener y almacenar la información leída de la plata telefónica.

- 3.1 Leer los datos enviados por el conmutador digital Infinity TX-2400L/XL.
- 3.2 Interpretar los valores de la información obtenida.
- 3.3 Almacenar la información clasificada en la base de datos del sistema.
- 4. Crear servicios Web para el acceso a la información almacenada en la base de datos.
  - 4.1 Mostrar llamadas recibidas
  - 4.2 Mostrar llamadas efectuadas
  - 4.3 Mostrar estado de los servicios.
- 5 Atención de peticiones
  - 5.1 Envío de secuencias de configuración al conmutador digital Infinity TX-2400L/XL para la configuración de servicios telefónicos.

### **2.3.3 Requisitos no funcionales.**

#### **2.3.3.1 Requisitos de Interfaz externa.**

La aplicación propuesta poseerá una interfaz sencilla y cómoda para el usuario a quien va dirigida, o sea el administrador de la central telefónica. Se realizará siguiendo el diseño de las ventanas de Windows, para mayor familiaridad con la misma.

#### **2.3.3.2 Requisitos de funcionalidad.**

Esta aplicación debe ser capaz de funcionar independiente del número de datos que se estén recibiendo o enviando desde o hacia la central.

#### **2.3.3.3 Requisitos de usabilidad.**

El administrador de la central telefónica, que es el único usuario de esta aplicación, no necesitará contar con una preparación previa para operar la misma pues la mayoría de las operaciones que realiza esta aplicación se hacen de forma automática. Las operaciones no automatizadas tampoco requerirán de la

intervención de un experto pues en su mayoría se trata de opciones de configuración y visualización de reportes fácilmente entendibles por cualquier persona.

#### **2.3.3.4 Requisitos de Rendimiento.**

La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen, la velocidad que se logre en la transferencia de información entre el conmutador digital Infinity TX-2400L/XL y la computadora donde estará ejecutándose esta aplicación y por la velocidad en la que se realicen las consultas a la base de datos.

La velocidad de ejecución de las operaciones de la aplicación debe ser la más alta posible, fundamentalmente en la lectura de la información que se recibe desde el puerto SMDR del conmutador digital. Esto se debe a la rapidez y al volumen de datos que se envía por este puerto ya que a través de él el conmutador notifica la ocurrencia de diversos sucesos asociados con las líneas telefónicas conectadas a la central.

#### **2.3.3.5 Requisitos de confiabilidad.**

Es precisa la ejecución perpetua de la aplicación para la actualización constante de la base de datos. Esta característica es fundamental debido que la gran cantidad de líneas telefónicas habilitadas actualmente en el conmutador Infinity de la UCI y las que están por habilitar hacen bastante probable que en cada segundo se genere por lo menos una o varias notificaciones de sucesos en el conmutador o una petición de solicitud de servicio por parte de los abonados. En caso de alguna falla que pudiera provocar la detención de la aplicación se debe proceder a su inmediata reparación de lo contrario se corre el riesgo de perder una inmensa cantidad de información de mucha importancia para los abonados.

### **2.3.3.6 Requisitos de diseño e implementación.**

Se desea construir una herramienta para la plataforma Windows, por lo que se debe escoger un lenguaje de programación y SGBD que permitan la ejecución de aplicaciones en esta plataforma. En este documento se proponen Microsoft SQL Server para la gestión de datos y C# para el desarrollo de la aplicación.

Para garantizar una mejor documentación del sistema, así como el uso de tecnología de avanzada, el análisis y el diseño se realiza usando UML (Unified Modeling Language). Como herramienta de apoyo a este Lenguaje de Modelación se utiliza el Rational Rose de la Suite de Rational por las facilidades que brinda en el modelado visual de UML.

### **2.3.3.7 Requisitos de software y de hardware.**

Para el correcto desempeño de esta aplicación es necesario que la misma se ejecute en una computadora que contemple las características que se describen en el anexo 2.

Se requiere que la computadora implicada en la comunicación con el conmutador digital Infinity TX-2400L/XL debe permitir comunicación utilizando el protocolo de transferencia TCP/IP y debe tener al menos uno de los puertos series RS-232C disponible para establecer la comunicación directamente con el puerto SMDR del conmutador Infinity TX-2400L/XL. La conexión TCP/IP se utilizará para comunicar esta computadora con el terminal de comunicación el cual puede estar ubicado en una localización remota. La computadora conectada directamente al conmutador digital Infinity debe tener el software TCP-Com u otro similar que permita realizar una transferencia de información entre una conexión TCP/IP en un puerto local con una conexión por uno de los puertos series locales o viceversa. El cable de conexión con el conmutador de cumplir las características descritas en el epígrafe 2.3.1 de este documento.

Debe estar instalada en la computadora donde se ejecuta el terminal de comunicación al menos la versión cliente de la aplicación SQL Server para que

esta pueda acceder a la base de datos donde se almacenará la información que se obtendrá a partir de los datos enviados por el conmutador digital Infinity.

### **2.3.3.8 Requerimientos legales.**

El software pertenece al proyecto UCI-Ciudad Digital, específicamente al área de Telecomunicaciones.

## **2.4 Especificación de los casos de uso**

### **2.4.1 Identificación de los actores**

**Administrador:** Es la persona que controla el correcto funcionamiento de la aplicación. Tiene la autoridad para ver el estado del sistema, configurar la herramienta, etcétera. Puede realizar las mismas operaciones que los abonados además de tomar decisiones que contribuyan una mejor prestación de servicio por parte de la central.

**Sistema operativo:** Se refiere al contador de tiempo del sistema operativo que es quien desencadena las acciones automáticas del terminal. Es un actor abstracto.

**Usuario:** Es la aplicación o la persona (abonado) que hará uso de la información almacenada en la base de datos del sistema o solicitará la configuración de uno de los servicios contemplados en este sistema.

## 2.4.2 Listado de los casos de usos del sistema

Tabla 2.1. Listado de casos de uso del sistema.

CU-1	Establecer conexión entre el conmutador Infinity TX-2400L/XL y la base de datos.
Actor	Administrador.
Descripción	Se establece la conexión entre el conmutador digital Infinity y la computadora donde se ejecutará el terminal de servicio y de ésta con la base de datos.
Referencia	RF 1.

CU-2	Agregar un teléfono.
Actor	Administrador.
Descripción	El administrador solicita agregar a la base de datos los datos de un teléfono instalado en una de las entradas libres de las líneas del conmutador.
Referencia	RF 2.

CU-3	Agregar tipo de cuenta.
Actor	Administrador.
Descripción	El administrador solicita agregar a la base de datos los datos de un nuevo tipo de cuenta telefónica.
Referencia	RF 2.

CU-4	Agregar servicio.
Actor	Administrador.
Descripción	El administrador solicita agregar a la base de datos los datos de un nuevo servicio telefónico.
Referencia	RF 2.

CU-5	Recepción automática de registros de datos.
Actor	Sistema operativo.
Descripción	Se realiza periódicamente y consisten en la recepción, interpretación y almacenamiento de los datos correspondientes a los mensajes enviados por el conmutador digital Infinity.
Referencia	RF 4.

CU-6	Configuración automática de servicios.
Actor	Sistema operativo.
Descripción	Se realiza periódicamente y consiste atención de las solicitudes de configuración de servicios contenida en la base de datos.
Referencia	RF 4.

CU-7	Habilitar/Cancelar servicio despertador automático.
Actor	Administrador, usuario.
Descripción	El administrador o un usuario solicitan activar o cancelar el servicio de despertador automático a un teléfono.
Referencia	RF 5.

CU-8	Habilitar/Cancelar servicio de no molestar.
Actor	Administrador, usuario.
Descripción	El administrador o el usuario solicitan activar o cancelar el servicio de no molestar a un teléfono.
Referencia	RF 5.

CU-9	Mostrar llamadas registradas por las líneas del conmutador.
Actor	Administrador.
Descripción	Se solicita un reporte de las llamadas que ha sido registrada por el conmutador digital. Se muestra un listado con las llamadas efectuadas.
Referencia	RF 4.

CU-10	Mostrar llamadas realizadas por un teléfono.
Actor	Administrador
Descripción	El administrador solicita visualizar las llamadas realizadas en un rango de fecha determinado. Se gestionan los registros en la base de datos del sistema mostrándose un listado de las llamadas recibidas en el intervalo de fecha especificado.
Referencia	RF 3.

CU-11	Mostrar llamadas recibidas por un teléfono.
Actor	Administrador
Descripción	El administrador solicita visualizar las llamadas recibidas en un rango de fecha. Se gestionan los registros en la base de datos del sistema mostrándose un listado de las llamadas recibidas en el intervalo de fecha especificado.
Referencia	RF 4.

CU-12	Reportes de estado de los teléfono.
Actor	Administrador
Descripción	El administrador solicita un reporte con información referente a un teléfono. El sistema gestiona la solicitud mostrándose el estado de configuración de los servicios antes mencionados.
Referencia	RF 4.

CU-13	Eliminar registros de llamadas.
Actor	Administrador.
Descripción	Se realiza con el objetivo de eliminar registros en desuso de la base de datos y liberar espacio en el servidor de datos.
Referencia	RF 2.

CU-14	Obtener reportes y estadísticas sobre llamadas.
Actor	Administrador.
Descripción	Se realiza con el objetivo de obtener distintos tipos de información relativa a los teléfonos, las cuentas y los servicios.
Referencia	RF 2.

### 2.4.3 Diagrama de casos de uso

Para una mejor comprensión de los casos de usos del sistema, así como para representar de forma más clara la relación entre los actores del mismo, se definieron dos paquetes de casos de uso:

- **Administración:** Contiene los casos de usos referentes a la opciones de administración del sistema así como a los de su configuración.
- **Solicitudes:** Contiene los casos de usos asociados con la tramitación de las peticiones de información de los usuarios así como de las solicitudes de configuración de servicios telefónicos.

Las figuras que a continuación se muestran dan una idea más exacta de los casos de uso contemplado en cada paquete:

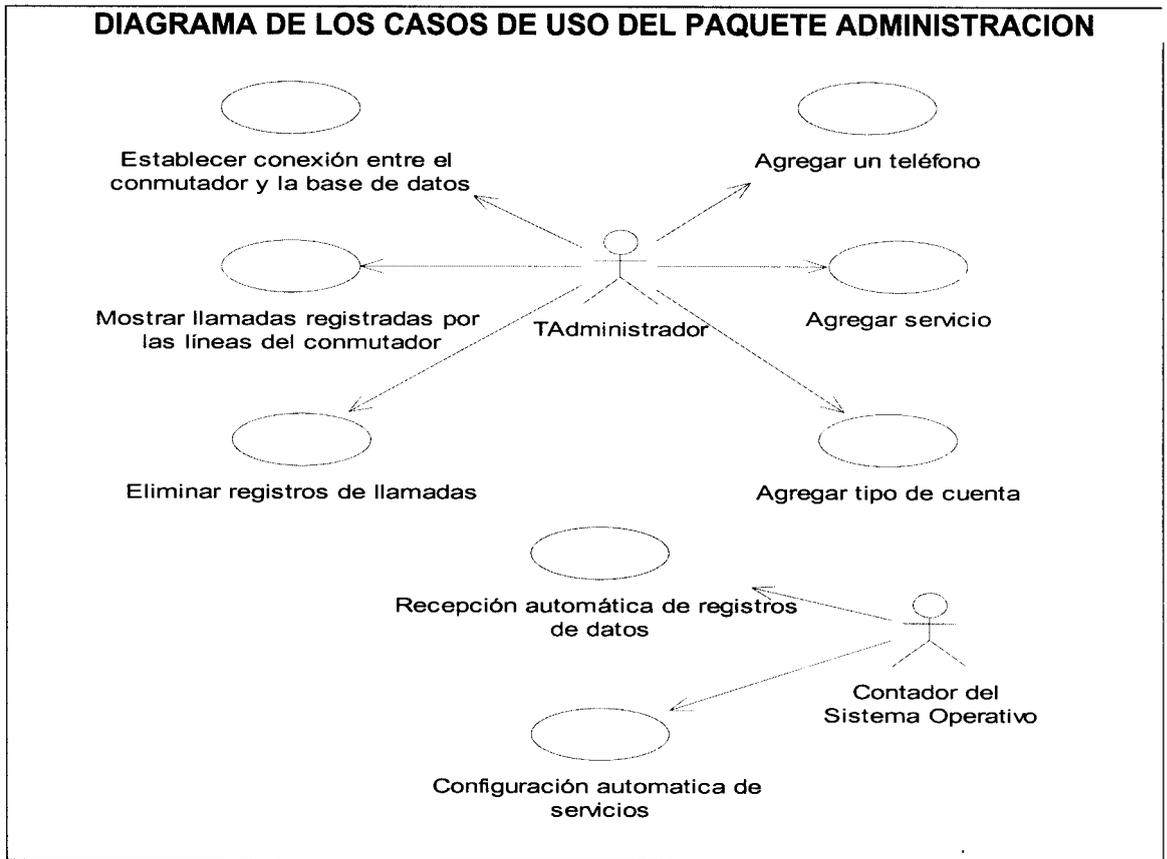


Figura 2.1 Casos de uso del paquete Administración

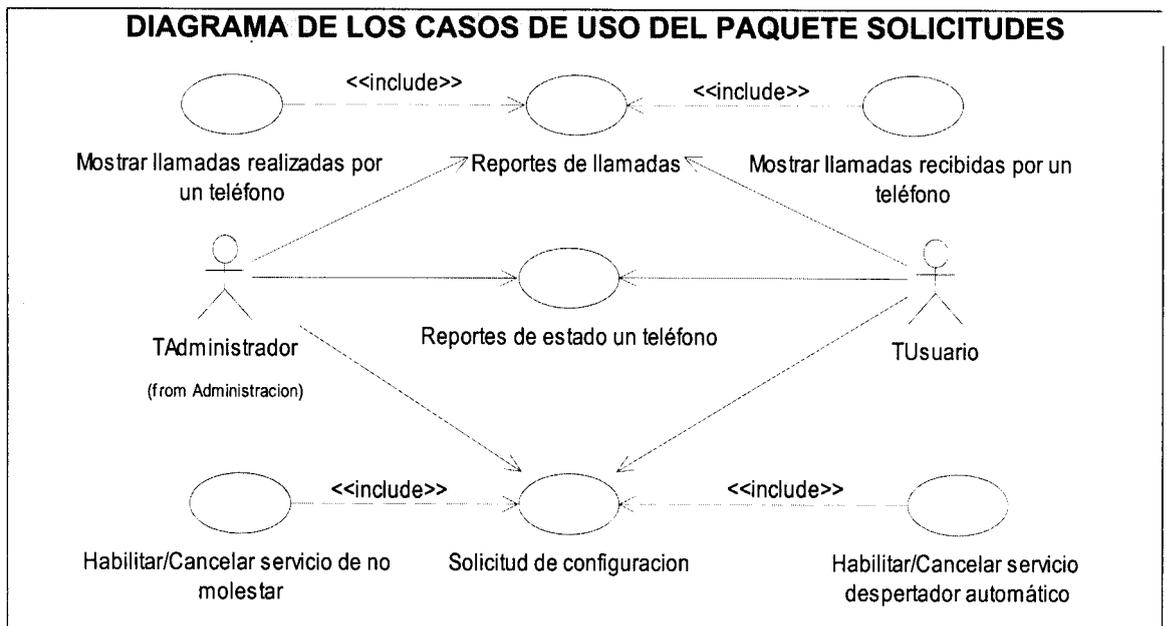


Figura 2.2 Casos de uso del paquete Solicitudes.

## 2.4.4 Casos de uso por ciclos.

Los ciclos de desarrollo se organizan en torno a los requerimientos de los casos de uso. Se asigna un ciclo para desarrollar uno o más casos, en dependencia de la complejidad que estos posean.

### 2.4.4.1 Casos de uso del núcleo central.

Para la selección de los casos que deben iniciar el primer ciclo de desarrollo o núcleo central es necesario identificar cuáles son los que influyen de manera determinante en la arquitectura básica del sistema.

**Tabla 2.2** Casos de uso núcleo central.

Código	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-1	Establecer conexión con servidor TCP/IP y el servidor de la base de datos.	Administración	<ul style="list-style-type: none"> <li>• Son los casos de uso de mayor complejidad y representan los procesos primarios que se requieren para el funcionamiento del sistema e influyen en la arquitectura básica.</li> <li>• Incluyen funciones urgentes.</li> </ul>
CU-2	Agregar un teléfono.		
CU-3	Agregar tipo de cuenta.		
CU-4	Agregar servicio.		
CU-5	Recepción automática de registros de datos.		
CU-6	Configuración automática de servicios		
CU-9	Mostrar llamadas registradas		
CU-12	Reportes de estado de teléfono.		
CU-13	Eliminar registros de llamadas.		

**Tabla 2.3** Casos de uso del segundo ciclo.

Código	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-7	Habilitar/Cancelar servicio despertador automático.	Solicitudes.	<ul style="list-style-type: none"> <li>• Constituyen casos de uso de importancia pero no imprescindible para el funcionamiento inicial del sistema.</li> <li>• La obtención de reportes y estadísticas acerca de las llamadas no constituyen procesos básicos.</li> <li>• La configuración inicial de la herramienta se hará manualmente.</li> </ul>
CU-8	Habilitar/Cancelar servicio de no molestar.		
CU-11	Mostrar llamadas recibidas por un teléfono.		
CU-14	Obtener reportes y estadísticas sobre llamadas.		

## 2.5 Modelo conceptual.

El modelo conceptual muestra los conceptos básicos del sistema, sus partes y relaciones. Se realiza a través de un diagrama de clases de UML simplificado, en el cual se representan las clases preliminares, las asociaciones preliminares entre las clases, y los atributos de las clases. Una cualidad esencial que debe ofrecer un modelo conceptual es que representa cosas del mundo real, no componentes del software [LARMAN 02]. Se representa en la figura 2.4.

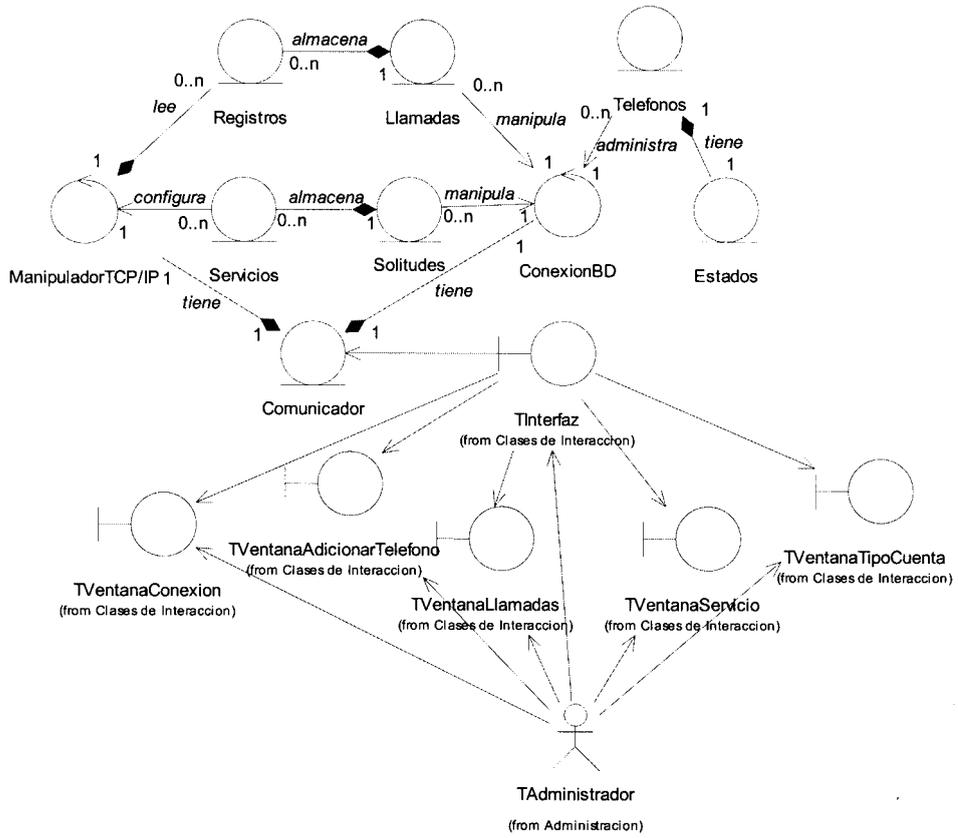


Figura 2.4 Modelo conceptual

# Capítulo 3

## Análisis y diseño del sistema.

### Introducción.

En el presente capítulo se representa la expansión de los casos de uso, el modelo conceptual, también se presentarán los diagramas de secuencias y la descripción de los contratos correspondientes a cada acción de dichos diagramas, los diagramas de colaboración así como el Diagrama de Clases y la Base de Datos, utilizando para el modelado el Lenguaje Unificado de Modelación (UML).

### 3.1 Extensión de los casos de uso

El anexo 3 muestra la expansión de cada uno de los casos de uso contemplados en esta etapa de implementación y que corresponden fundamentalmente al primer ciclo de desarrollo o sea a los casos de uso del núcleo central. Se incluyen además algunos casos de uso del ciclo con el objetivo de mostrar algunos datos de interés para el administrador de la central telefónica.

### 3.2 Diagramas de interacción de los casos de uso

Los diagramas de interacción describen la forma en que cada operación detectada en los diagramas de secuencia lleva a cabo sus responsabilidades y modifica el estado del sistema, mostrando el modo en que los objetos interactúan a través de mensajes. En UML los diagramas de interacción pueden representarse a través de los Diagramas de Colaboración y/o de los Diagramas de Secuencia. Los Diagramas de Secuencia muestran interacciones entre objetos basadas en el tiempo [RUP 03].

El anexo 4 muestran los diagramas de clases de diseño y los diagramas de interacción para los casos de uso que componen el primer ciclo de desarrollo de

este sistema. En el anexo 5 se muestra la descripción de cada una de las clases del modelo de diseño

### 3.3 Diagrama de clases.

El diagrama de clases es una descripción de los modelos de objetos, contiene clases y las relaciones estructurales y de herencia existentes entre ellas. Este se obtiene como resultado del refinamiento del modelo conceptual y de los diagramas de secuencia. La definición de clase incluye definiciones para atributos y responsabilidades. La figura 3.1 muestra el diagrama de clases.

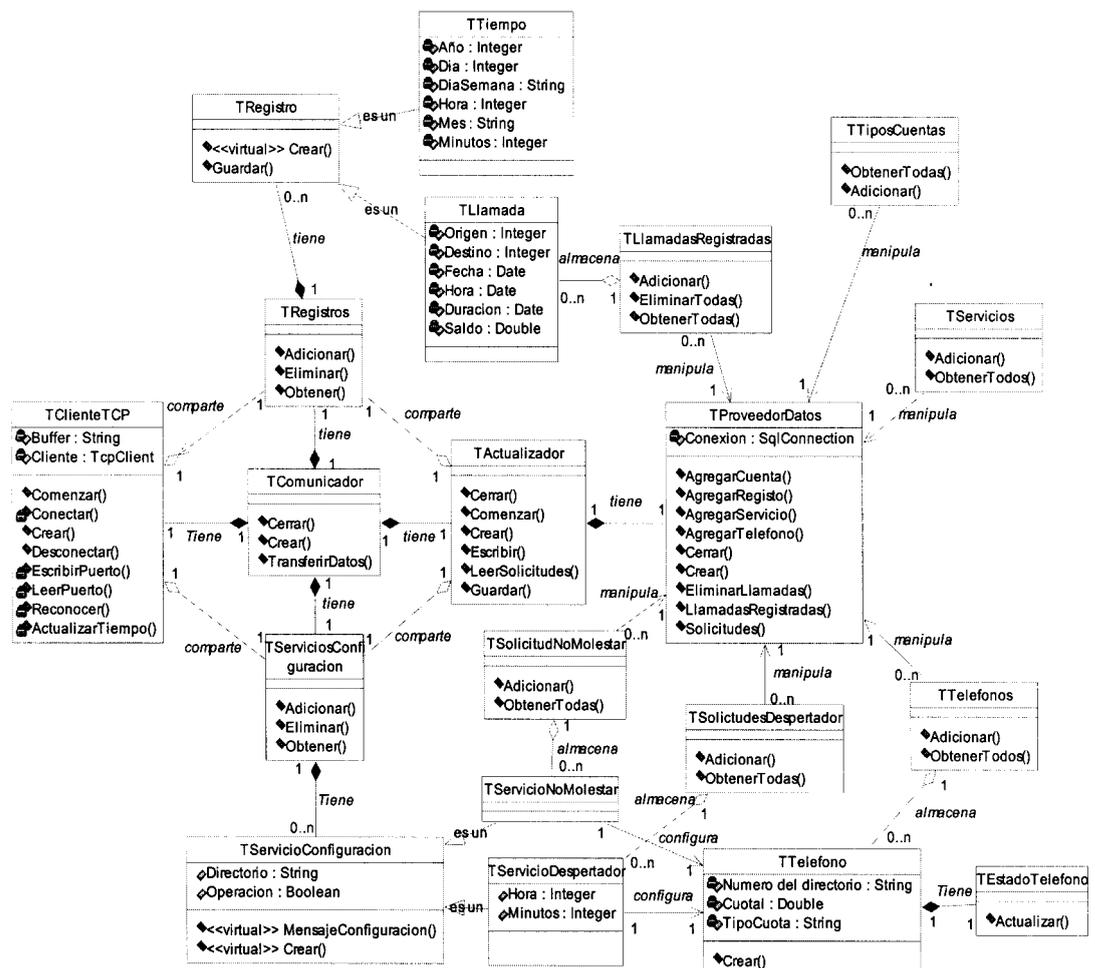


Figura 3.1 Diagrama de clases.

## 3.4 Diseño de la base de datos

Con el objetivo de lograr la persistencia de los datos y que estos puedan ser consultados en cualquier momento por cualquier usuario se utilizó, en la concepción de este prototipo, una base de datos relacional. El uso de esta técnica de almacenamiento está muy difundido actualmente debido a las garantías que ofrece en cuanto a la durabilidad de los datos se refiere.

El anexo 6 muestra la descripción de las tablas concebidas inicialmente para este prototipo y en el anexo 7 se muestran los diseños lógicos y físicos de la base de datos respectivamente utilizada en la versión inicial de este prototipo.

# Capítulo 4

## El TAC-24K

### Introducción.

Este capítulo tiene como objetivo brindar una descripción general del proceso de desempeño del terminal de comunicación propuesto en este documento para el conmutador digital Infinity TX-2400L/XL. En él se explica además como utilizando la programación con múltiples hilos de ejecución se logra evitar la pérdida de información mientras esta se lee, se clasifica y se almacena. Como un último punto del capítulo se detallan cada uno de los mensajes de comunicación que utiliza TAC-24K para intercambiar información con el conmutador digital.

### 4.1 Descripción general de TAC-24K

TAC-24K es una aplicación de usuario para la plataforma Windows, desarrollado en el lenguaje de programación C# del paquete de Visual Studio .Net edición del 2003 con el objetivo de facilitar la gestión de la central telefónica de conmutación digital Infinity TX-2400L/XL de TX-Communication. Entre las características de esta aplicación se encuentran:

- Comunicación a través de la conexión TCP/IP con un servidor de esta conexión que está a su vez conectado por medio de uno de sus puerto serie (RS-232C) con el conmutador digital Infinity TX-2400L/XL lo cual posibilita la interacción entre las partes involucradas en la comunicación.
- Ejecución de un grupo de tareas de manera automática, entre ellas:
  - Actualización de una base de datos con correspondiente traducción de la información derivada de la comunicación con el conmutador digital.

- Habilitación y/o cancelación de algunos servicios telefónicos concebidos para este tipo de conmutador e indicados en la base de datos.
- Interfaz gráfica para la interacción con el usuario de la misma (administrador) la cual le permite realizar de forma cómoda y sencilla:
  - La puesta en marcha de la aplicación.
  - La visualización de reportes acerca de las líneas telefónicas de la central.
  - La habilitación y/o cancelación de servicios telefónicos para una línea en específico.

Como se describió en capítulos anteriores se requieren de ciertas características en el hardware de la PC que ejecute el terminal lo cual posibilita su óptimo desempeño. El anexo 2 muestra estas principales características.

## **4.2 Comunicación con el conmutador**

La comunicación con el conmutador Infinity TX-2400L/XL es posible realizarla a través de una conexión utilizando el protocolo de transferencia de datos TCP/IP. Los parámetros para la conexión contemplan los siguientes atributos:

- Nombre del servidor de la conexión.
- Puerto habilitado para la conexión.

Una vez establecidos los parámetros de conexión pueden enviarse y recibirse desde y hacia el conmutador, respectivamente, los mensajes de comunicación en el formato descrito en el subepígrafe 1.1.6 de este documento. Los mensajes enviados desde el Terminal de comunicación hacia el conmutador Infinity son interpretados por este último el cual lleva a cabo la operación contenida en el mensaje. Los mensajes enviados desde el conmutador Infinity al terminal de comunicaciones contienen información relativa a los sucesos y estados de las

líneas telefónicas. De esta forma se puede operar con algunas funcionalidades de una central digital de este tipo. La comunicación a través de estos tipos de mensajes hace posible prescindir de la interacción manual con ella salvo en caso de averías o de instalación de nuevas facilidades. Por tanto el objetivo fundamental del TAC-24K es precisamente lograr una comunicación eficiente con el conmutador digital Infinity, pues de ella depende el correcto funcionamiento de las demás utilidades de ésta aplicación.

La comunicación TCP/IP del TAC-24K con el conmutador digital Infinity se realiza a través de un objeto *NetworkStream*, el cual constituye el controlador que conecta el hardware de la computadora con el Sistema Operativo para la transferencia de datos. Este objeto incluido en el paquete de clases del espacio de nombre *System.Net.Sockets* de Visual Estudio .Net y se emplea para transmitir y recibir datos mediante el protocolo de comunicación TCP/IP. Entre las propiedades y funciones más importante de este objeto se encuentran:

- **DataAvailable:** es una propiedad que indica si hay datos disponibles en el objeto para ser leídos.
- **Read:** es una función que permite obtener los datos que pueden ser leídos del objeto.
- **Write:** es una función que permite escribir enviar datos a través de la conexión establecida por el objeto.

### 4.3 Base de datos del sistema

Como se mencionó anteriormente los mensajes de comunicación entre TAC-24K y el conmutador digital Infinity poseen un formato o estructura poco amigable y entendible a primera vista. Es por ello que una de las funcionalidades de TAC-24K es precisamente la traducción de los valores contenidos en estos mensajes a información clara y entendible por los usuarios para luego almacenarlos en una

base de datos, desde la cual, posteriormente, pueden ser leídos y entendidos por cualquiera que los solicite.

La base de datos diseñada para almacenar la información recogida e interpretada por TAC-24K tiene la estructura lógica y física propuesta en el anexo 7.

Una de las tablas más importante en este diseño es la que se encarga de almacenar la información correspondiente a las llamadas que se han registrado en el conmutador digital. En los registros de esta tabla se encuentran datos sobre el teléfono que realiza la llamada, el teléfono que la recibe, la fecha, hora y minutos de la llamada, así como el costo o saldo de la misma.

Otras tablas no menos importantes son las que almacenan las solicitudes de configuración de servicios telefónicos por parte de los abonados, así por ejemplo una de ellas está destinada a almacenar las solicitudes de configuración del servicio de matutino o despertador y otra para las del servicio de no molestar o bloqueo de llamadas. La primera recoge información del número de teléfono para el cual se solicita el servicio, la hora y los minutos para la cual lo solicita y si lo quiere es habilitarlo o cancelarlo. La segunda contiene, además del número del teléfono para el que solicita el servicio, si lo desea habilitar o cancelar.

La base de datos fue creada y diseñada en el sistema de gestión de bases de datos relacionales *Microsoft SQL Server* y cuenta con el sistema de seguridad que este es capaz de brindar con el objetivo de restringir el acceso de los usuarios a la misma.

## **4.4 Interfaz gráfica para la ejecución de las tareas**

Otro de los objetivos trazados por este terminal es facilitar el proceso de comunicación entre el especialista encargado de la operación y la central de conmutación, por lo cual se ha creado una interfaz gráfica al estilo de las ventanas de Windows para la gestión de las distintas operaciones contempladas por el terminal. El objetivo primario de esta decisión fue el de concebir una aplicación con interfaz amigable y cómoda a diferencia de las que se pueden utilizar

actualmente debido a que éstas últimas, en su mayoría, están concebidas para la interpretación de comandos al estilos de MS-DOS. De esta forma se produciría la comunicación a un nivel más elevado, pues el especialista no tendría que pensar ya en las secuencias de bytes que debe enviar para ejecutar una operación determinada, sino que ahora indica la operación y TAC-24K se encarga del proceso a más bajo nivel.

## 4.5 Las tareas automáticas.

TAC-24K como el sistema automático que en realidad es tiene como principal responsabilidad la actualización automática de una base de datos y la configuración automática de algunos servicios proveídos por el conmutador digital Infinity TX-2400L/XL y solicitados por los abonados de éste. Para llevar a cabo estas actividades fue necesario auxiliarse de mecanismos de ejecución concurrente de procesos así como de mecanismos de comunicación entre los mismo. La presencia de estos mecanismos se debe fundamentalmente a la diferencia de velocidades de comunicación entre el conmutador digital Infinity TX-2400L/XL y TAC-24K y entre éste último y el gestor de datos de SQL Server de Microsoft.

Para evitar la pérdida de los datos enviados por el conmutador Infinity hacia TAC-24K se utiliza un hilo de ejecución encargado de recoger e interpretar dicha información. Una vez que la información se ha analizado y se han separados sus partes se almacenan de forma temporal en una estructura interna en forma de lista la cual es recorrida por otro hilo que se encarga de agregarla poco a poco en la base de datos del terminal.

De la misma forma que fue descrito el procedimiento anterior se realiza el proceso de configuración de servicios telefónicos solo que esta vez en orden inverso. Un hilo de ejecución se encarga de tramitar la información relativa a la configuración de los distintos servicios, convierte la misma en el objeto de servicio correspondiente al tipo de solicitud y los coloca en una estructura interna desde la

cual son recogidos posteriormente por otro hilo que es el encargado de enviar el correspondiente mensaje de configuración en dependencia del tipo de servicio al conmutador digital Infinity.

## 4.6 Las actividades concurrentes

Como se ha mencionado anteriormente el terminal de comunicación utiliza otros hilos además del hilo principal de ejecución para realizar las operaciones de lectura y actualización de la información y los servicios. Estos hilos secundarios comparten recursos, o sea listas de registros y servicios de configuración, por lo que el uso de este mecanismo de comunicación presupone el acceso a datos comunes. Este tipo de acceso concurrente trae aparejado problemas de sincronización por lo que para solucionarlo fue necesario auxiliarse de la clase `Mutex` del ámbito `System.Threading` de Visual Studio .Net con el objetivo de lograr el acceso exclusivo a los datos compartidos.

La clase `Mutex` contiene los métodos `WaitOne` y `ReleaseMutex` que pueden utilizarse para sincronizar procesos concurrentes que acceden a recursos compartidos. Dichos métodos se encargan respectivamente de bloquear y desbloquear los procesos que tratan de acceder a los recursos compartidos si estos están siendo utilizados o liberados respectivamente por uno de ellos.

## 4.7 Mensajes recibidos y servicios configurables por TAC-24K.

Existen un conjunto de servicios que pueden activarse o cancelarse para las distintas líneas telefónicas del conmutador digital Infinity TX-2400L/XL. El TAC-24K está concebido en su versión inicial para configurar solo algunos de ellos.

Debido a que no es posible contar con un teléfono equipado con todos los accesorios necesarios para obtener, por ejemplo, el servicio de identificación de llamada se hizo necesario almacenar esta información para de esta forma proveer a los abonados de una vía para identificar el origen de estas llamadas. Otros

servicios configurables desde TAC-24K son el servicio de despertador automático y el servicio de no molestar.

### 4.7.1 Identificación y facturación de llamadas.

Para darle al abonado la posibilidad de identificar las llamadas realizadas hacia su teléfono así como la facturación de las que se realizan desde su teléfono se almacenan en la base de datos del sistema TAC-24K los registros de llamadas que el conmutador Infinity envía a través del puerto SMDR del panel de control del conmutador.

El formato del mensaje correspondiente a los registros de llamadas se describe como sigue:

El registro de la fecha es enviado primero que el registro de los detalles de la llamada.

byte 1	STX	Comienzo del mensaje
byte 2,3	"30"	Código de la función
byte 4,5		El día
byte 6	"-"	
byte 7,8		El mes
byte 9	"-"	
byte 10,13		El año
byte 82	MSN	
byte 83,84	BCC	Bloque de chequeo de caracteres el cual es un OR exclusive de todos el texto en bytes.
byte 85	ETX	Fin del mensaje

Un registro detallado de una llamada consta de varios campos tales como el directorio de origen de la llamada, los dígitos marcados, la hora de comienzo de la llamada y la duración de la misma.

- |            |  |
|------------|--|
| byte 1     | STX (2) Comienzo del mensaje.  |
| byte 2,3   | "32" Código de la función.   |
| byte 4-8   | Número del directorio del teléfono que realiza la llamada. Note que el número del directorio puede contener 5 dígitos de longitud. La computadora debe poder despojar el primer dígito del número del directorio para formar un número de cuatro de 4 dígitos. |
| byte 9     | " " Espacio en blanco.   |
| byte 10,11 | Número del arrendatario del teléfono. El número del arrendatario es el grupo a que el teléfono pertenece. Ésta también puede ser una designación para el idioma del grupo.   |
| byte 12,13 | " " 2 Espacios en blancos  |
| byte 14-17 | El código de la ruta de acceso al tronco que realiza la salida de la llamada.  |
| byte 18    | " " Espacio en banco.  |
| byte 19-34 | Los dígitos marcados por la estación. Los dígitos marcados comienzan en el byte 19 y se completan con espacios en blancos si son menos de 16.  |
| byte 35    | " " Espacio en blanco.   |
| byte 36-47 | Código de autorización o código de cuenta. El código de autorización comienza con un número de 3 dígitos y continúa con 36 bytes de espacios en blancos. El código de cuenta consiste en 12 dígitos de espacio en blanco.                                      |

byte 48	" "	Espacio en blanco
byte 49,50		Hora de comienzo de la llamada (00-23).
byte 51	":"	Dos puntos
byte 52,53		Minutos de comienzo de la llamada (00-59).
byte 54	" "	Espacio en blanco
byte 55,56		Duración de la llamada - Horas.
byte 57	":"	Dos puntos
byte 58,59		Duración de la llamada - Minutos
byte 60	":"	Dos puntos
byte 61,62		Duración de la llamada - Segundos
byte 63	" "	Espacio en blanco.
byte 64-69		Número de pulsos recibidos desde la Oficina Central o calculado por el sistema.
byte 70	" "	Espacio en blanco.
byte 71-76		Cargo calculado para la llamada. El cargo es calculado multiplicando el número de pulsos medidos durante la llamada por el costo de un pulso configurado por el administrador.
byte 77	" "	Espacio en blanco.
byte 78-81		El número del directorio del tronco asociado con la llamada.
byte 82	" "	Espacio en blanco si la llamada no fue transferida
	"*"	Si la llamada fue transferida a otro teléfono.
byte 82		MSN
byte 83,84	BCC	Bloque de chequeo de caracteres el cual es un OR exclusive de todos el texto en bytes.

byte 85      ETX    Fin del mensaje

## 4.7.2 Servicio de despertador automático

El servicio despertador automático o matutino como se le conoce comúnmente es el servicio que le permite programar al sistema para que el teléfono del abonado que lo solicita haga la función de despertador en el momento que el abonado lo desee. Para habilitar o cancelar este servicio desde TAC-24K se utiliza un mensaje que contiene la información del directorio que lo solicita, si lo desea habilitar o cancelar y para el primer caso el tiempo para el cual lo desea configurar.

El formato del mensaje enviado desde TAC-24K para habilitar la llamada del servicio de despertador automático es el siguiente:

byte 1	STX	Comienzo del mensaje.
byte 2,3	F1, F2	Código de la función.
byte 4,5	P1, P2	Código del proceso.
byte 6,7,8,9	D1, D2, D3, D4.	
byte 10,11	H1, H2.	
byte 12,13	M1, M2.	
byte 14	MSN	
byte 16	BCC	Bloque de chequeo de caracteres el cual es un OR exclusive de todos el texto en bytes.
byte 15	ETX	Fin del mensaje

F – Código de la función

15      Despertador Automático.

P – Código del proceso

00      Cancelar el despertador automático.

01      Habilitar el despertador automático.

D – Número del directorio.

H - Hora para la llamada del despertador automático.

M – Minutos para la llamada del despertador automático.

Note que el tiempo del despertador automático es H1H2:M1M2

Para todos los mensajes el MSN - número de sucesión del mensaje = 0', 1',..., 9' /0x30-0x39 / - si el mensaje no se repite el MSN = MSN + 1. MSN puede ser verificado si es requerido por el SMDR. Es válido sólo para los mensajes del conmutador al Terminal. Para los mensajes del Terminal al conmutador puede ponerse el MSN a '0' o ignorarlo.

### 4.7.3 Servicio de no molestar

Este servicio no permite que la estación de timbre en un momento en el que el abonado no desea ser molestado. Para habilitar o cancelar este servicio desde TAC-24K se utiliza un mensaje que contiene información del directorio que solicita el servicio y si lo que desea es habilitarlo o cancelarlo. El formato de este mensaje es como sigue:

byte 1            STX    Comienzo del mensaje

byte 2,3        F1, F2 Código de la función.

byte 4,5        P1, P2        Código del proceso.

byte 6,7,8,9    D1,D2,D3,D4

byte 10,11    BCC    Bloque de chequeo de caracteres el cual es un OR  
exclusive de todos el texto en bytes.

byte 12        ETX    Fin del mensaje

F – Código de la función

16    No molestar.

P - Código del proceso

- 00 Deshabilitar la opción de no molestar.
- 01 Habilitar la opción de no molestar.

D – Número del directorio.

#### 4.7.4 Mensaje de difusión.

Los mensajes de difusión constituyen avisos generados por el conmutador digital en el momento mismo que estos se generan. Entre los mensajes podemos encontrar por ejemplo: aviso de actualización de la hora, aviso de teléfono descolgado, aviso de teléfono timbrando, etcétera. Todos los sucesos ocurridos en las líneas telefónicas que de alguna manera puedan generar el envío de un pulso eléctrico al conmutador digital son registrados por éste y notificados correspondientemente por medio de un mensaje por el puerto SMDR. Estos mensajes reciben el nombre de mensajes de difusión debido a que precisamente transmiten información relacionada con los sucesos del conmutador. Los mensajes de difusión son identificados por el código de función 99 seguido del tipo de información que el sistema quiere indicar. Los valores contenidos en los bytes del mensaje están expresados en el sistema de notación hexadecimal por lo que deben ser interpretados haciendo su traducción al sistema decimal. El formato general es el siguiente:

byte 1	STX	Comienzo del mensaje
byte 2,3	99	Código de la función.
byte 4,5	P1,P2	Tipo de mensaje.
byte 6,n		Datos para el tipo de mensaje
byte n+1	ETX	Fin del mensaje

Algunos tipos de mensajes de difusión.

Tipo	Código
Línea descolgada	0x01
Línea colgada	0x02
Actualización de tiempo	0x76

#### 4.7.4.1 La hora desde el conmutador digital.

El conmutador digital emite un mensaje de difusión o mensaje de sistema cada vez que registra el transcurso de un minuto. Para indicar la fecha y hora que tiene configura el conmutador digital éste envía un mensaje con el siguiente formato:

byte 1	STX	Comienzo del mensaje
byte 2,3	99	Código de la función.
byte 4,5	76	Actualización de tiempo.
byte 6,7		Hora configurada en el conmutador (0 a 23).
byte 8,9		Minutos configurados en el conmutador (0 a 59).
byte 10,11		Día configurado en el conmutador (1 a 31).
byte 12,13		Mes configurado en el conmutador (1 a 12).
byte 14,15		Año configurado en el conmutador (desplazamiento + 1980).
byte 16,17		Día de la semana configurado en el conmutador (0 a 6).
byte 18-23		Número del registro de llamada.
byte 18	ETX	Fin del mensaje

## Conclusiones.

---

A pesar del corto período de prueba al que fue sometido TAC-24K, las corridas iniciales del mismo arrojaron como resultado que se pudieran cumplir los objetivos propuestos en este documento. TAC-24K ha hecho posible que la UCI pueda contar, de manera rápida y eficiente, con información relacionada con los teléfonos de su central telefónica, la cual, anterior a esto, era difícil o casi imposible de obtener por los abonados de su central telefónica.

Además de esto TAC-24K es capaz de almacenar en una base de datos los datos referentes a las llamadas registradas por el conmutador digital Infinity y permite configurar los servicios de despertador automático y de no molestar para cualquier teléfono conectado al conmutador digital Infinity de la UCI. Esta aplicación automatiza el proceso de actualización de la referida base de datos y agiliza el proceso de gestión de la información. El programa permite, de forma rápida y sencilla, la configuración de los servicios telefónicos concebidos en su versión inicial casi de manera instantánea, posibilitando de esta manera que los abonados lo puedan configurar desde cualquier punto de la universidad siempre y cuando puedan hacer llegar su solicitud a la base de datos del TAC-24K.

Hasta el momento solo es posible contar con información referente a las llamadas internas y de las que se realizan desde los teléfonos internos hacia los externos, ya que estas son los únicos tipos de llamadas que estuvieron configuradas en el conmutador digital Infinity de la UCI para ser notificadas por él durante la etapa de desarrollo del terminal propuesto en este trabajo. A pesar de esta limitante el terminal queda preparado para almacenar también la información referente a las llamadas que se hacen desde el exterior hacia las líneas internas en caso de que estas comiencen a notificarse.

El estudio inicial de las potencialidades del conmutador digital Infinity TX-2400L/XL, de las utilidades que éste brinda al enviar información sobre los sucesos que ocurren en el mismo y de la posibilidad de realizar algunas configuraciones de servicios telefónicos, todo esto a través de la interfaz de

grabación de detalles de mensajes, permite concluir que muchas son operaciones y la información que pueden realizarse u obtenerse al interactuar con el referido conmutador a través de esta interfaz.

Todas las actividades desarrolladas este prototipo de terminal fueron llevadas a término prácticamente sin la tutoría de un personal acreditado en la materia ya que en el centro no existe tal personal. A pesar de todo eso se obtuvieron resultados relevantes con esta versión inicial por lo que este prototipo de terminal de comunicación constituye, sin lugar a dudas, el punto de partida para el desarrollo futuro de un software que sea capaz de brindar otras funcionalidades no implementadas en esta versión inicial. Así por ejemplo, pudiera pensarse en la posibilidad de que dicho software permita configurar otros servicios como el enrutamiento de llamadas, el retorno automático de llamadas, etcétera.

## **Recomendaciones.**

---

Con el objetivo de mejorar y concluir una versión más completa del prototipo de Terminal de comunicación propuesto en este documento recomendamos lo siguiente:

- Estudiar los diferentes mensajes de difusión enviados por el conmutador digital Infinity con el fin de aumentar la información que se pueda brindar referentes a las líneas telefónicas.
- Completar los casos de usos del segundo ciclo.
- Implementar la ayuda del prototipo.
- Considerar la posibilidad de crear una tabla en la base de datos con el objetivo de mantener un historial de las llamadas registradas y de agilizar el proceso de búsqueda de información referente a dichas llamadas.
- Implementar un mecanismo para permitir que los abonados puedan configurar más de un servicio de despertador a un mismo teléfono utilizando la hora del conmutador.
- Agregar al terminal una vía de comunicación directa con el conmutador digital en la cual no intervenga otra PC que no sea la que ejecuta el terminal, o sea, que este debe contar con un mecanismo de transferencia de información a través de un puerto serie (RS-232C) de esta PC conectado directamente al puerto del conmutador digital.

## Referencias bibliográficas.

---

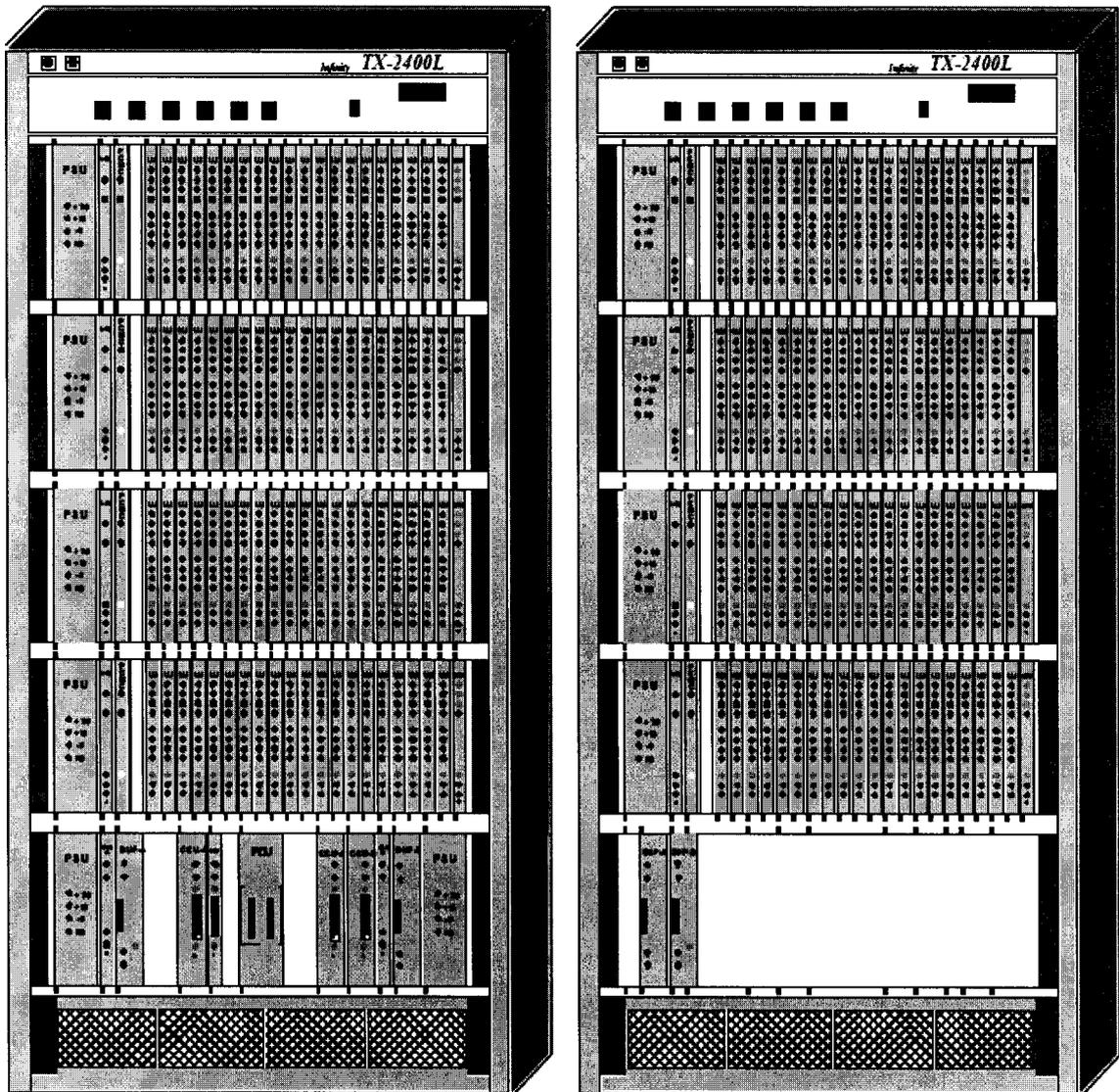
- [TX/Communication Canadá Inc] *Infinity TX-2400L /XL*. Digital Switching System. General Description.
- [MATOS 99] – Matos, Rosa María. *Introducción al trabajo con Base de Datos*. Asignatura de Sistemas de Gestión de Base de Datos, 1999.
- [SQL] – SQL <http://www.arsys.es/soporte/programacion>
- [UML 03] - Tutorial de UML.
- [UCI 03] – *Documento acerca de la Proyección estratégica de la UCI*. UCI, Ciudad de la Habana, 2003.
- [MACHADO 03] - Machado, Alejandro G. Documento acerca del Proyecto UCI Ciudad Digital. UCI, Ciudad de la Habana, 2003.
- [LARMAN 02] – Larman, Craig. *UML y patrones, introducción al análisis y diseño orientado a objetos*. Segunda edición, Prentice-Hall, 2002.
- [ALVAREZ 00] - Alvarez, Sofía, Hernández Anaisa. *Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML*. La Habana, 2000.
- [RUP 03] - Rational Unified Process. Rational Software Corporation. “*Rational Unified Process*”. Version 2001A.04.00, Copyright 1987-2001.

## **Bibliografía**

---

- [1] TX/Communications Canada Inc. Application notes.
- [2] TX/Communications Canada Inc. Feature Implementation Guide.
- [3] TX/Communications Canada Inc. Insatallation and Maintenace Manual.

# Anexo 1. TX-2400L/XL con dos gabinetes.



## Anexo 2. Requerimiento de hardware para el desempeño de la aplicación.

Procesador	Procesador Pentium III a 600 MHz o superior.
Sistema operativo	Windows 2000, Windows XP o cualquier otra versión donde este instalado el Framework de .Net y la herramienta de desarrollo Visual Estudio .Net
Disco	Se requiere de uno con suficiente capacidad en la maquina que almacenará la información debido al gran volumen de la misma. De la capacidad del disco dependerá el tiempo de persistencia de los datos en la base de datos.
Puerto	Se requiere de al menos un puerto RS-232C
Cable de conexión	Cable de conexión con conector estándar EIA-9 para puerto RS-232C (9-pin) y conector estándar EIA-25 para el puerto SMDR del conmutador (25-pin)

### Anexo 3. Extensión de casos de uso implementados.

Caso de uso		Actor
CU-1	Establecer conexión entre el conmutador Infinity TX-2400L/XL y la base de datos.	Administrador
<p><b>Propósito:</b> Establecer una conexión entre el conmutador digital Infinity TX-2400L/XL y la base de datos del sistema.</p>		
<p><b>Resumen:</b> El administrador del sistema establece los parámetros de conexión con el conmutador digital y la ubicación de la base de datos en el servidor de datos.</p>		
Acción del actor	Respuesta del sistema	
<p>1. Solicita establecer la conexión entre el conmutador digital y la base de datos.</p> <p>3. Entra los datos necesarios para esta conexión indicando nombre y puerto del servidor TCP/IP y el nombre del servidor de datos y el de la base de datos.</p>	<p>2. Muestran los campos necesarios para la recepción del servidor de conexión TCP/IP y el puerto habilitado para dicha conexión y el servidor de datos y la base de datos.</p> <p>4. Almacena los datos de la conexión.</p> <p>5. Crea la conexión solicitada entre las partes involucradas y almacena la conexión establecida.</p> <p>6. Notifica el estado de conexión establecida.</p>	
<b>Flujo Alternativo</b>		
<b>Paso 1.</b> Si desea cerrar la conexión establecida.		
<b>Paso 2:</b> Cierra las conexiones con el servidor TCP/IP y con el de datos.		
<b>Paso 4:</b> Notifica el estado de conexión no establecida.		

Caso de uso		Actor
CU-2	Agregar un teléfono.	Administrador
<p><b>Propósito:</b> Agregar a la base de datos los información relativa a una nueva línea telefónica instalada en el conmutador.</p>		
<p><b>Resumen:</b> El administrador del sistema solicita agregar la información referente a un teléfono que incluye su número de directorio, tipo de cuenta y monto inicial de la misma.</p>		
<p><b>Casos de uso relacionados:</b> CU-1</p>		
Acción del actor		Respuesta del sistema
<p>1. El administrador solicita agregar la información de una nueva línea telefónica.</p> <p>3. Especifica el número de teléfono o directorio, el tipo de cuenta y el monto inicial de la misma.</p>		<p>2. Solita el número del nuevo teléfono o directorio, el tipo de cuenta y el monte inicial de la misma.</p> <p>4. Agrega en la base de datos del sistema la información de la nueva línea.</p>
<p><b>Flujo alternativo:</b></p> <p><b>Paso 4.</b> Devuelve estado de error en la inserción.</p>		

Caso de uso		Actor
CU-3	Agregar tipo de cuenta.	Administrador
<p><b>Propósito:</b> Agregar a la base de datos los información relativa a un nuevo tipo de cuenta telefónica.</p>		
<p><b>Resumen:</b> El administrador del sistema solicita agregar la información referente a un nuevo tipo de cuenta telefónica que incluye su nombre, descripción y cantidad con que inicia.</p>		
<p><b>Casos de uso relacionados:</b> CU-1</p>		
Acción del actor		Respuesta del sistema
<p>1. El administrador solicita agregar la información de un nuevo tipo de cuenta telefónica.</p>		<p>2. Solita el nombre, la descripción y la cantidad con que inicia de la cuenta.</p> <p>4. Agrega en la base de datos del</p>

3. Especifica el nombre, la descripción y la cantidad de dinero con que se inicia la cuenta.	sistema la información de la nueva cuenta telefónica.
<b>Flujo alternativo:</b>	
<b>Paso 4.</b> Devuelve estado de error en la inserción.	

Caso de uso		Actor
CU-4	Agregar Servicio.	Administrador
<b>Propósito:</b> Agregar a la base de datos los información relativa a un nuevo servicio telefónico.		
<b>Resumen:</b> El administrador del sistema solicita agregar la información referente a un nuevo servicio telefónico que incluye su nombre, tipo y descripción.		
<b>Casos de uso relacionados:</b> CU-1		
Acción del actor		Respuesta del sistema
1. El administrador solicita agregar la información de un nuevo servicio telefónico.		2. Solita el nombre, el tipo y la descripción del servicio.
3. Especifica el nombre, el tipo y la descripción del servicio.		4. Agrega en la base de datos del sistema la información del nuevo servicio telefónico.
<b>Flujo alternativo:</b>		
<b>Paso 4.</b> Devuelve estado de error en la inserción.		

Caso de uso		Actor
CU-5	Recepción automática de registros de datos.	Sistema operativo
<b>Propósito:</b> Actualizar en la base de datos del sistema las tablas de las llamadas o el estado de los teléfonos.		
<b>Resumen:</b> Es una operación automática que se ejecuta periódicamente con el objetivo de leer e interpretar los mensajes enviados por el conmutador digital y almacenarlos sus datos en dependencia de la información que estos contengan.		

Casos de uso relacionados: CU-1	
Acción del actor	Respuesta del sistema
	<ol style="list-style-type: none"> <li>1. Establece la conexión entre las partes involucradas en la comunicación, dígame el conmutador digital, el Terminal y la base de datos del sistema.</li> <li>2. Un hilo de ejecución lee los mensajes del conmutador disponibles en la conexión TCP/IP.</li> <li>3. Interpreta los valores contenidos en los mensajes.</li> <li>4. Crea los objetos registros correspondientes y los almacena temporalmente en una lista de registros.</li> <li>5. Otro hilo de ejecución va eliminando registros de la lista y almacena la información contenida en los mismos en la base de datos.</li> </ol>

Caso de uso		Actor
CU-6	Configuración automática servicios.	Sistema operativo
<p><b>Propósito:</b> Configurar los servicios a partir de las solicitudes en la base de datos.</p>		
<p><b>Resumen:</b> Es una operación automática que se ejecuta periódicamente con el objetivo de enviar al conmutador digital los mensajes correspondiente a la configuración de determinados servicios telefónicos dándole atención a las solicitudes de los abonados.</p>		
Casos de uso relacionados: CU-1		
Acción del actor	Respuesta del sistema	
	<ol style="list-style-type: none"> <li>1. Establece la conexión entre las partes involucradas en la comunicación, dígame el conmutador</li> </ol>	

	<p>digital, el Terminal y la base de datos del sistema.</p> <p>2. Un hilo de ejecución lee las solicitudes de configuración de servicios almacenadas en la base de datos.</p> <p>3. Crea los objetos de servicios de configuración correspondientes y los almacena temporalmente en una lista de servicios de configuración.</p> <p>4. Otro hilo de ejecución va eliminando los servicios de la lista y obteniendo de estos los mensajes de configuración correspondiente para enviarlos al conmutador.</p>
--	---

Caso de uso		Actor
CU-9	Mostrar llamadas registradas por las líneas del conmutador.	Administrador
<p><b>Propósito:</b> Mostrar los datos referentes a las llamadas realizadas por las líneas telefónicas del conmutador digital Infinity TX-2400L/XL.</p>		
<p><b>Resumen:</b> El administrador del sistema solicita información referente a las llamadas que se han efectuado desde o hacia las líneas telefónicas del conmutador digital.</p>		
<p><b>Casos de uso relacionados:</b> CU-1</p>		
Acción del actor	Respuesta del sistema	
<p>1. Comienza cuando el administrador solicita información relacionada con las llamadas efectuadas o recibidas por las líneas telefónicas del conmutador.</p>	<p>2. Obtiene de la base de datos la información concerniente a las llamadas efectuadas o recibidas por las líneas telefónicas en rango de fecha especificado.</p> <p>3. Muestra el reporte con la información solicitada. En éste está contenido el origen y destino de la llamada así como la fecha, hora e importe de la misma.</p>	

Caso de uso		Actor
CU-10	Mostrar llamadas realizadas por un teléfono.	Administrador
<p><b>Propósito:</b> Obtener un reporte con datos referentes a las llamadas realizadas por un teléfono.</p>		
<p><b>Resumen:</b> Visualiza un reporte con información referente a las llamadas que ha efectuado un teléfono. Como resultado se obtiene un listado con la información solicitada.</p>		
<p><b>Casos de uso relacionados:</b> CU-1</p>		
Acción del actor		Respuesta del sistema
<ol style="list-style-type: none"> <li>1. Solicita información relacionada con las llamadas efectuadas por un teléfono determinado.</li> <li>3. Se especifican el número del teléfono del cual se desean obtener le reporte.</li> </ol>		<ol style="list-style-type: none"> <li>2. Solicita el número del teléfono o directorio del cual se desea obtener el reporte.</li> <li>4. Obtiene de la base de datos la información concerniente a las llamadas efectuadas por el teléfono indicado.</li> <li>5. Devuelve un reporte de datos con la información solicitada. En éste reporte está contenido el destino de las llamadas así como la fecha, hora e importe de las mismas.</li> </ol>

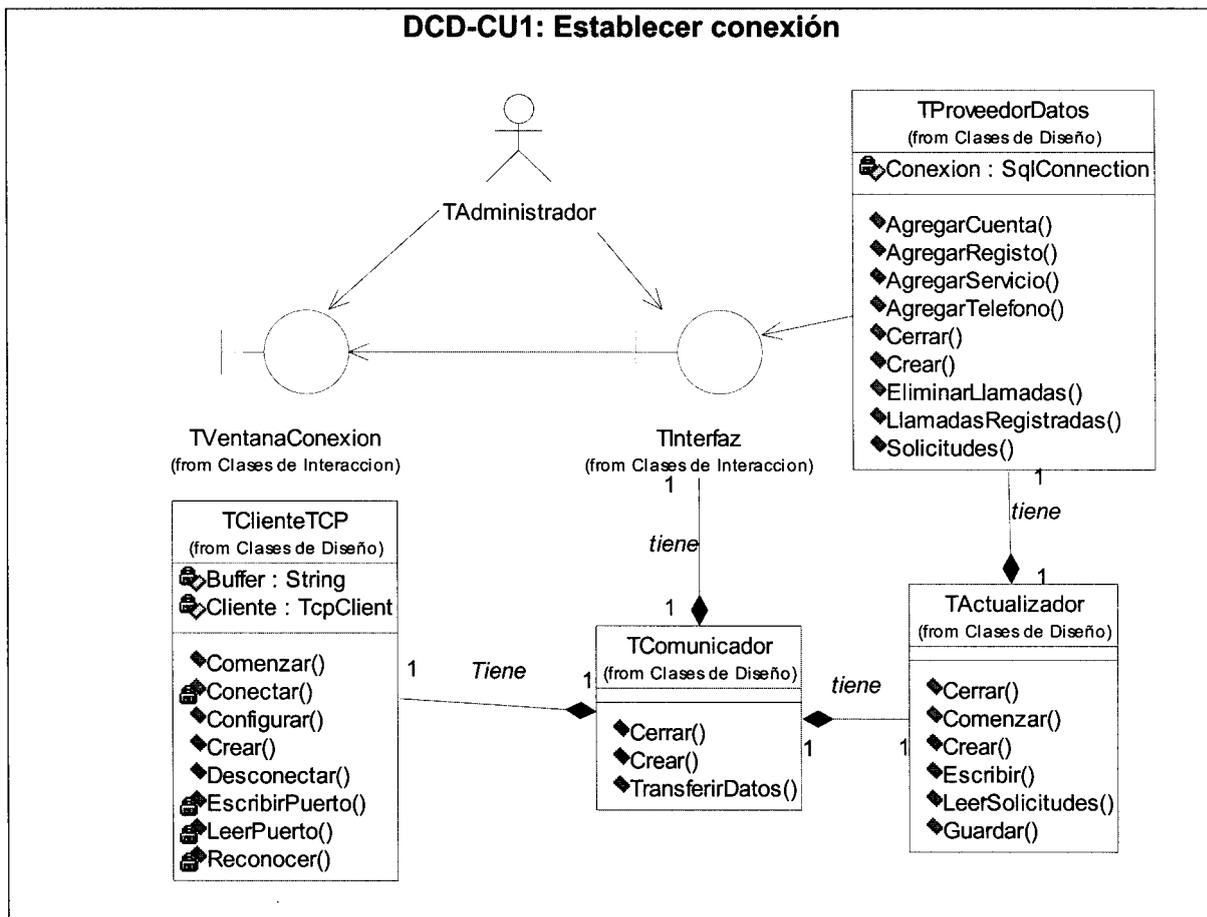
Caso de uso		Actor
CU-11	Mostrar llamadas recibidas por un teléfono.	Administrador
<p><b>Propósito:</b> Visualizar un reporte con datos referentes a las llamadas recibidas por un teléfono en un rango de fecha. Como resultado se obtiene un listado con la información solicitada.</p>		
<p><b>Resumen:</b> El administrador del sistema solicita información referente a las llamadas que ha recibido un teléfono. Como resultado se obtiene un reporte con la información solicitada.</p>		
<p><b>Casos de uso relacionados:</b> CU-1</p>		
Acción del actor		Respuesta del sistema
<ol style="list-style-type: none"> <li>1. Solicita información relacionada con las llamadas recibidas por un</li> </ol>		<ol style="list-style-type: none"> <li>2. Solicita el número del teléfono o directorio del cual se desea obtener el</li> </ol>

teléfono determinado. <b>3.</b> Especifica el número del teléfono del cual se desea obtener le reporte y el rango de fecha.	<p>reporte y el rango de fecha para el mismo.</p> <p><b>2.</b> Obtiene de la base de datos la información concerniente a las llamadas recibidas por el teléfono indicado en el rango dado.</p> <p><b>5.</b> Devuelve un reporte de datos con la información solicitada. En éste reporte está contenido el origen de las llamadas así como la fecha, hora e importe de las mismas.</p>
--	---

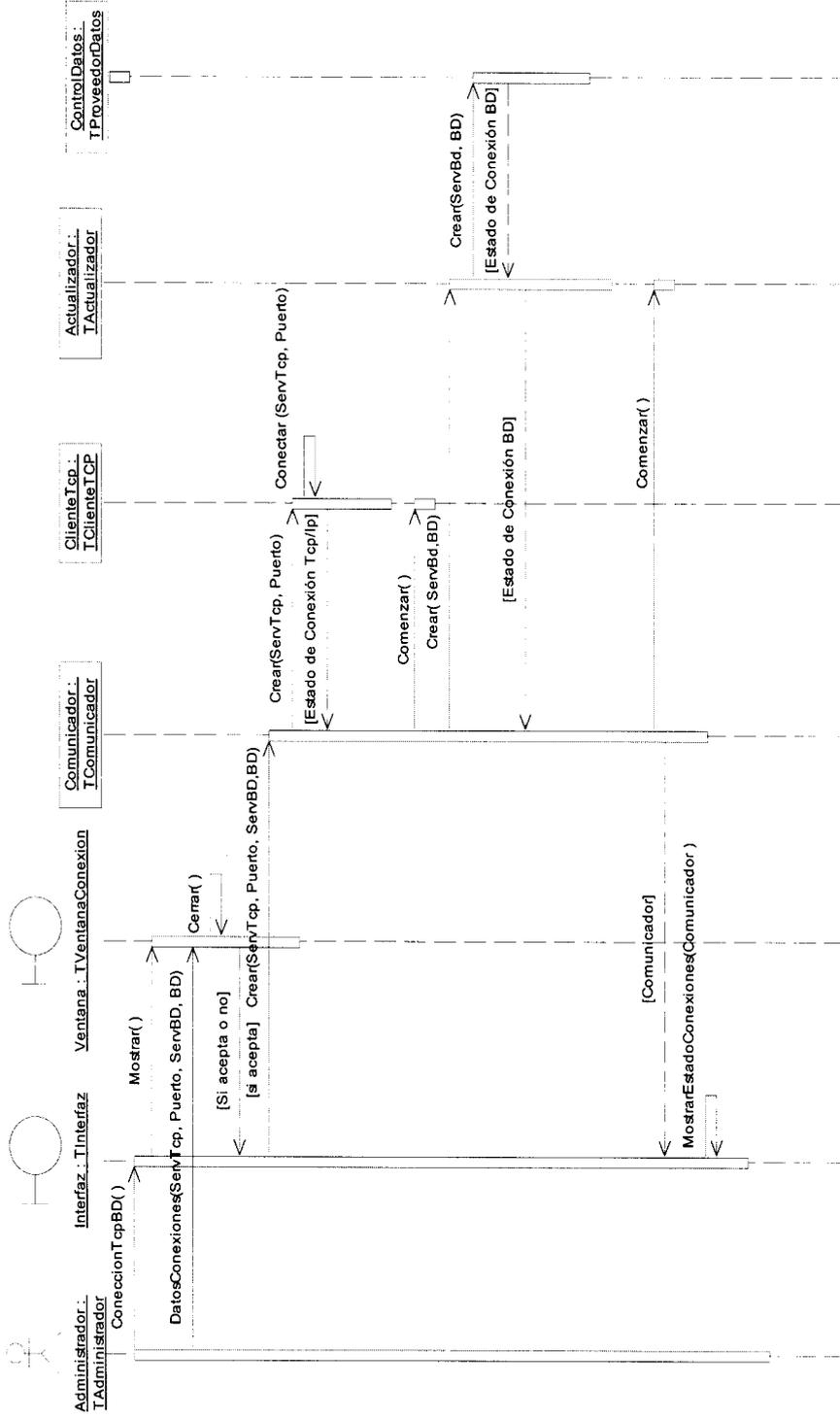
<b>Caso de uso</b>		<b>Actor</b>
CU-12	Reporte de estado de un teléfono.	Administrador
<b>Propósito:</b> Reportes de estado de los teléfonos.		
<b>Resumen:</b> El administrador del sistema solicita un reporte con la información del estado de los servicios de los teléfonos. Como resultado obtiene un reporte con la información solicitada.		
<b>Casos de uso relacionados:</b> CU-1		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	
<b>1.</b> Solicita obtener información sobre el estado de configuración de los servicios para las líneas del conmutador así como el estado de las líneas.	<p><b>2.</b> Obtiene de la base de datos la información solicitada.</p> <p><b>3.</b> Devuelve un reporte con la información obtenida la cual contiene datos sobre el estado del servicio de despertador automático (habilitado o cancelado), la hora y minutos del mismo así como el estado del servicio de no molestar (habilitado o cancelado)</p>	

<b>Caso de uso</b>		<b>Actor</b>
CU-13	Eliminar registros de llamadas.	Administrador
<b>Propósito:</b> Dar mantenimiento de la base de datos.		
<b>Resumen:</b> El administrador solicita dar mantenimiento a la base de datos y para ello elimina de la misma los registros de llamadas que llevan un tiempo determinado almacenado y que ya no son necesarios.		
<b>Casos de uso relacionados:</b> CU-1		
<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1. Solicita eliminar los registros de las llamadas efectuadas que se han almacenado en la base de datos.		2. Elimina de la base de datos los registros que cumplan la condición establecida por el límite de fecha.  3. Devuelve el estado de la eliminación.

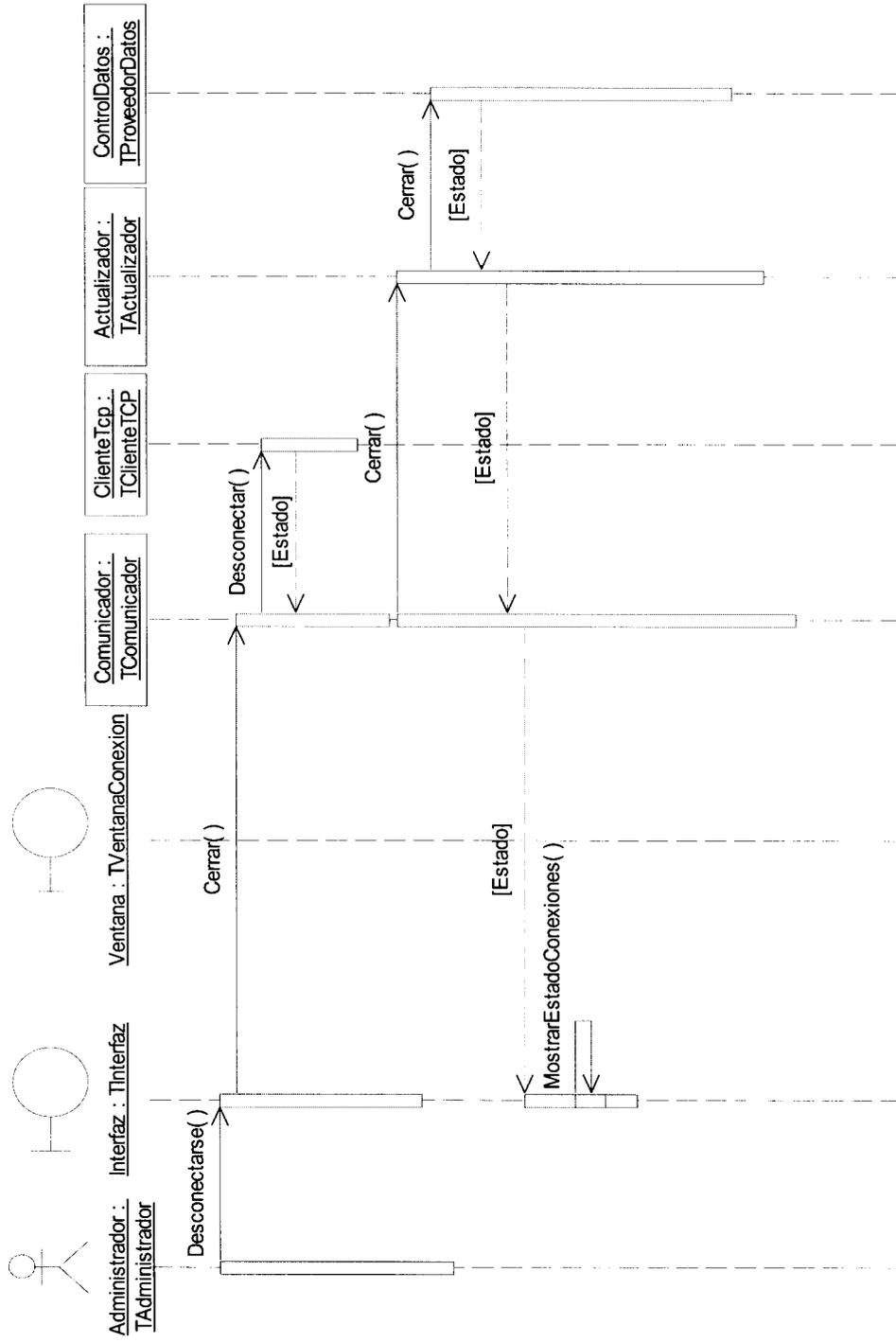
## Anexo 4. Diagramas de clases e interacción.



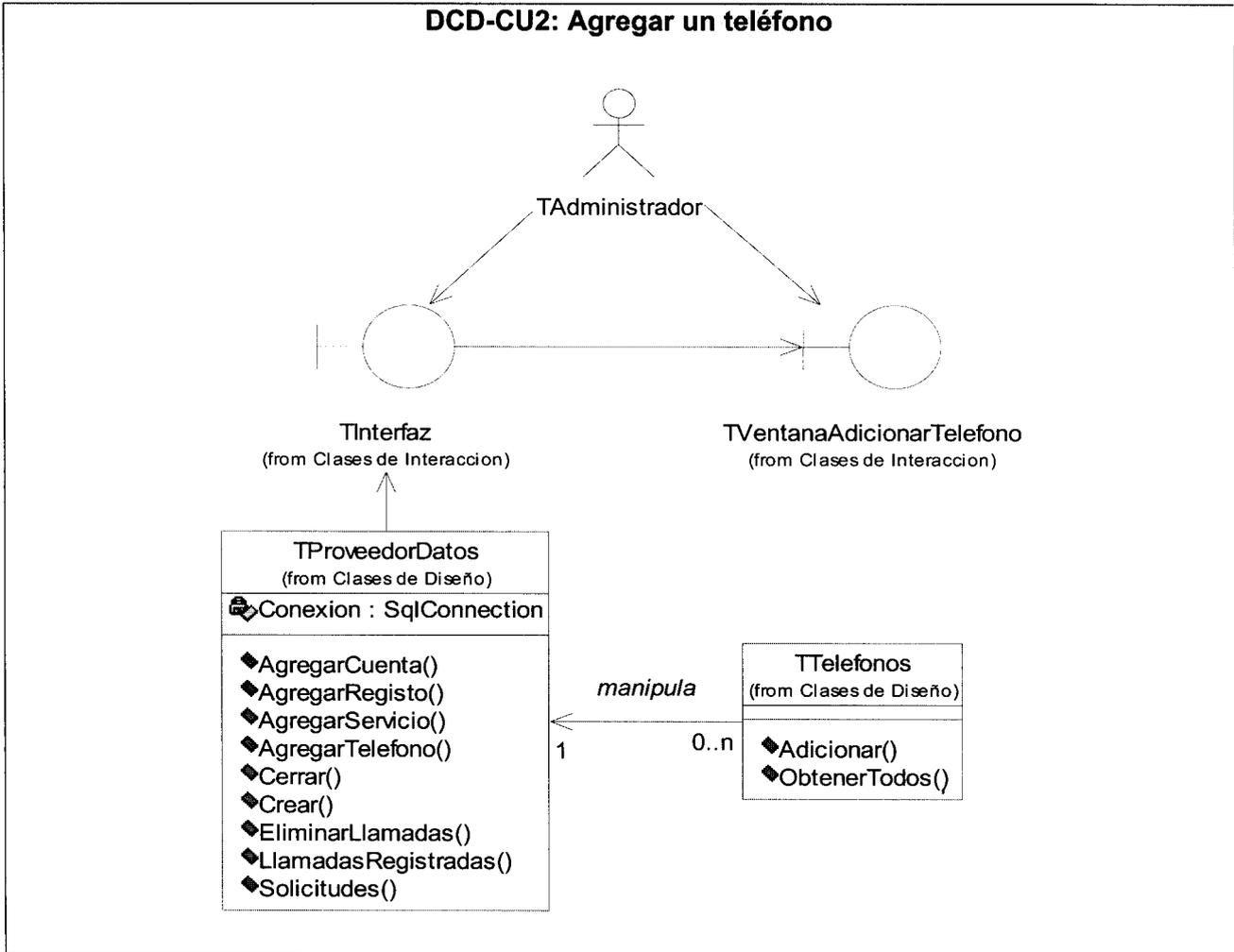
### DI-CU1: Establecer conexión (Abrir)



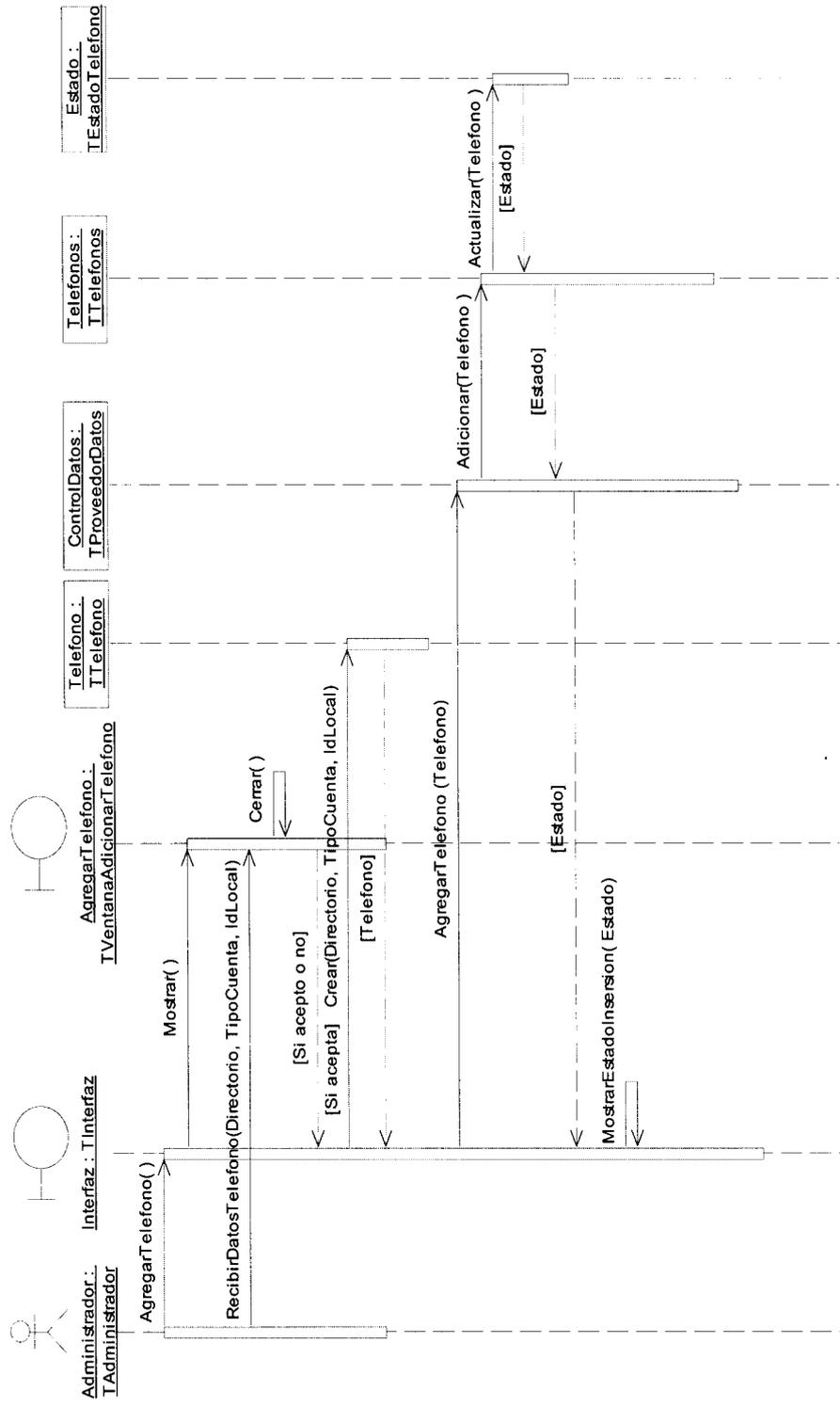
### DI-CU1: Establecer conexión (Cerrar)



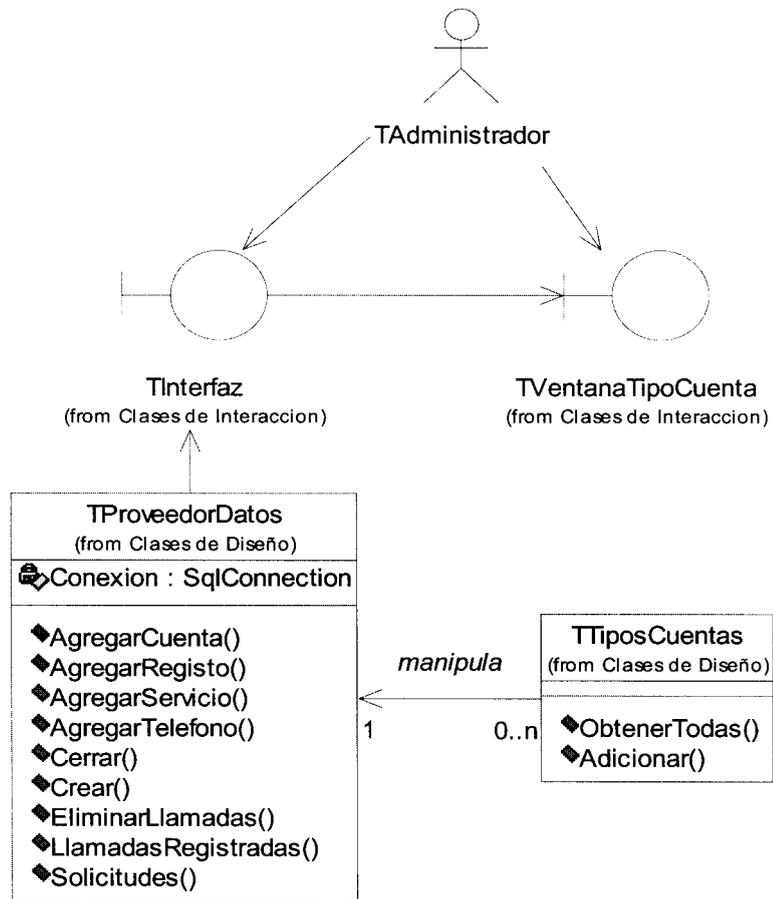
**DCD-CU2: Agregar un teléfono**



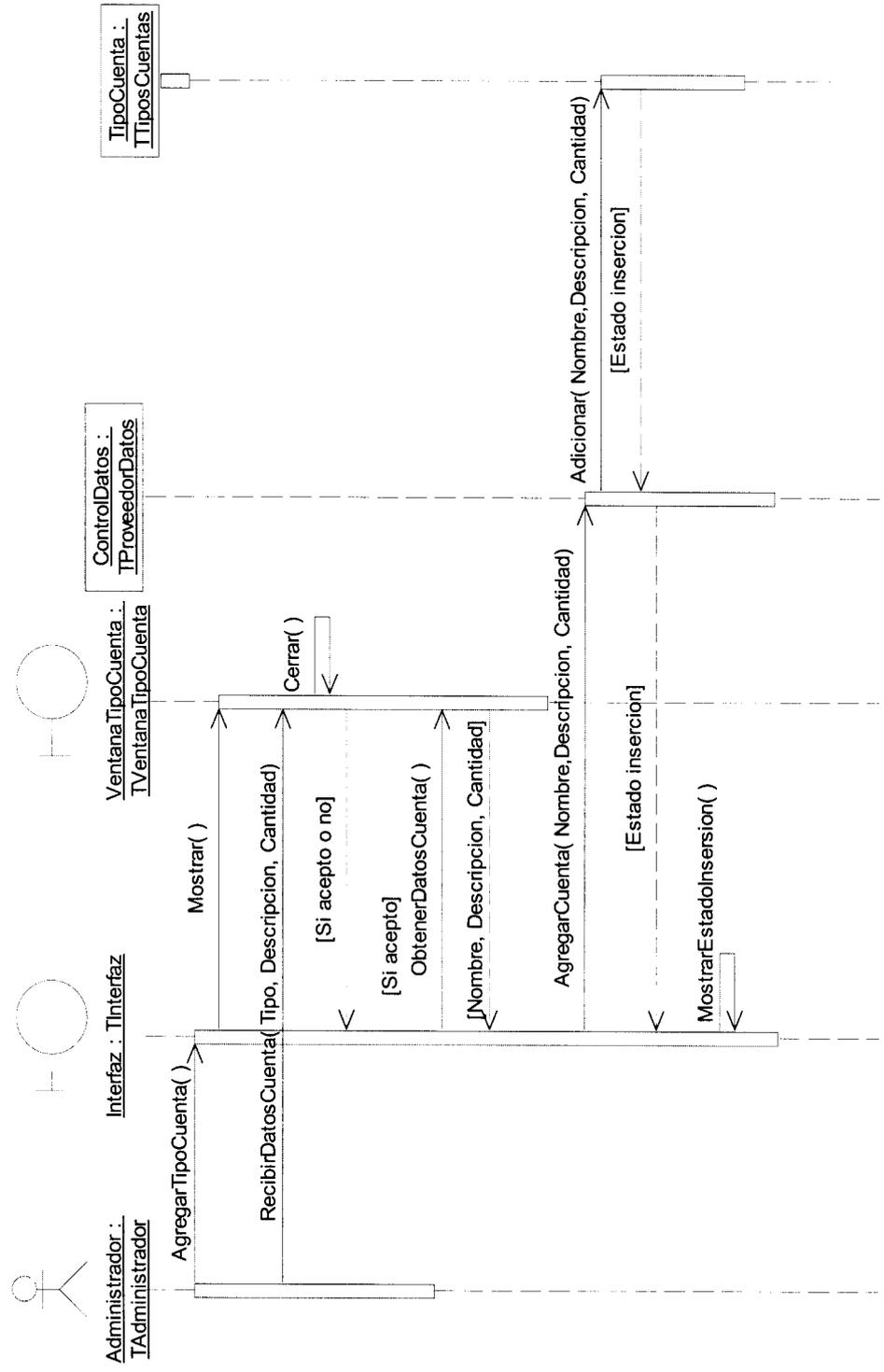
DI-CU2: Agregar un teléfono



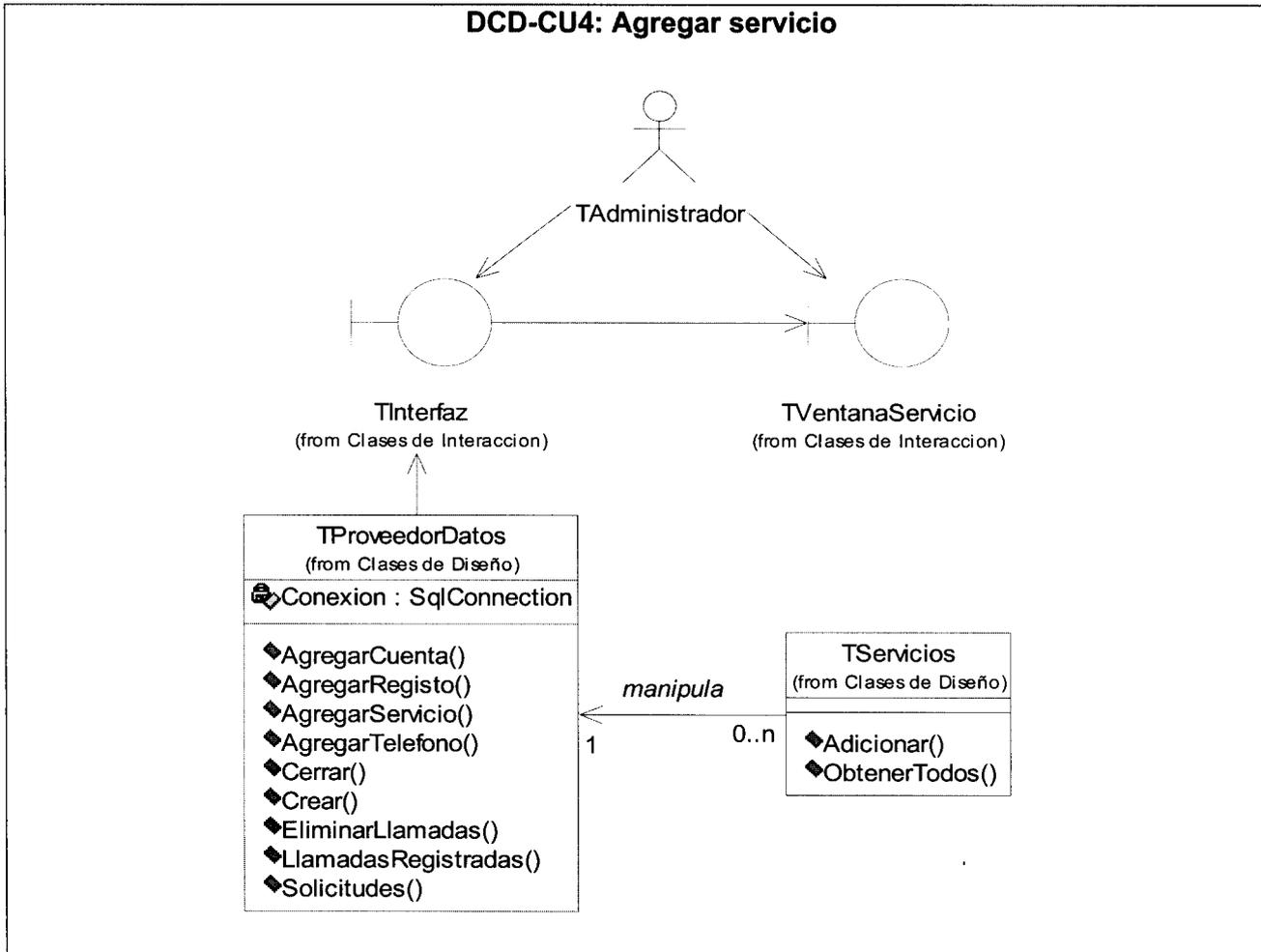
**DCD-CU3: Agregar tipo de cuenta**



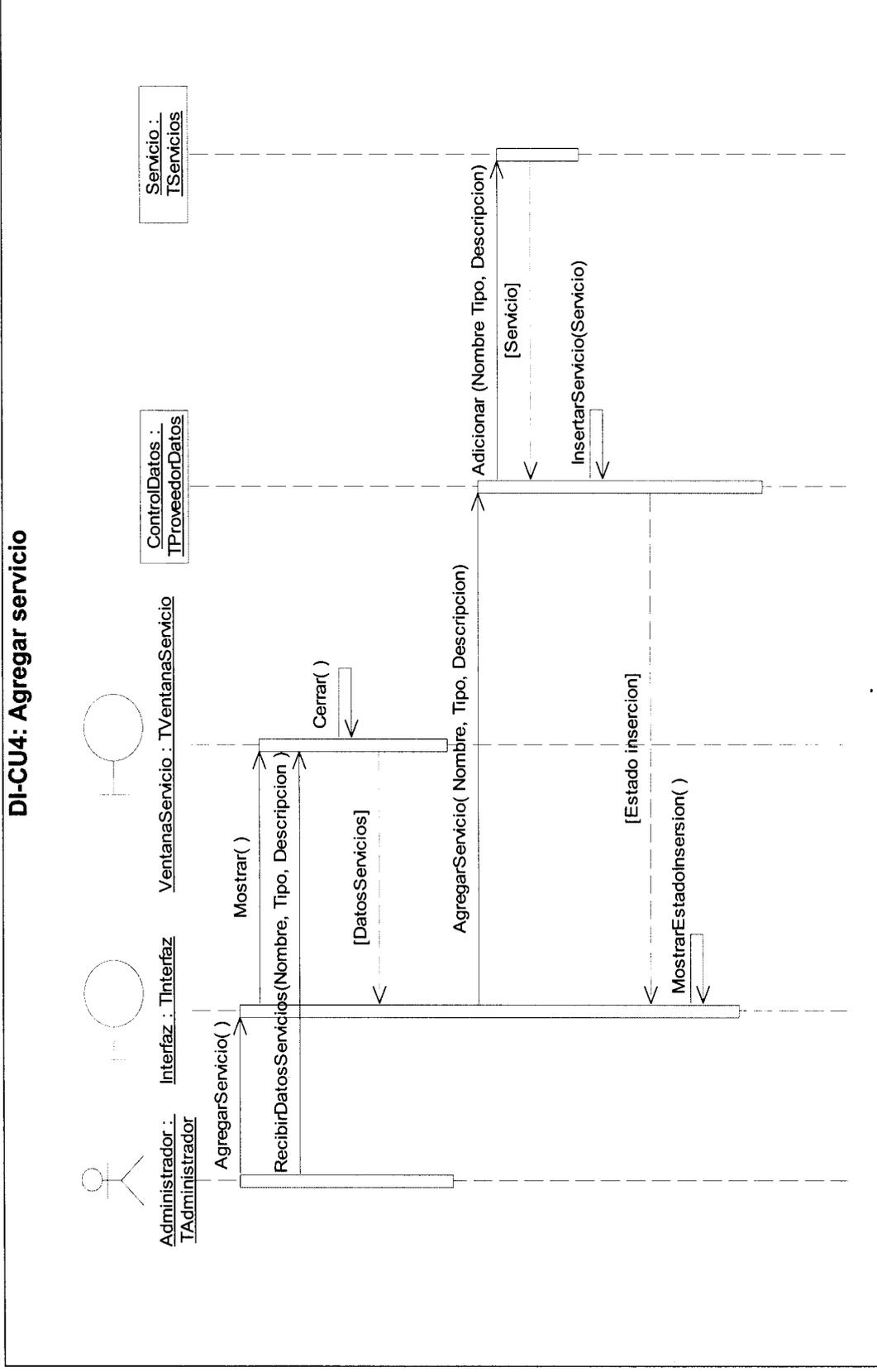
### DI-CU3: Agregar tipo de cuenta



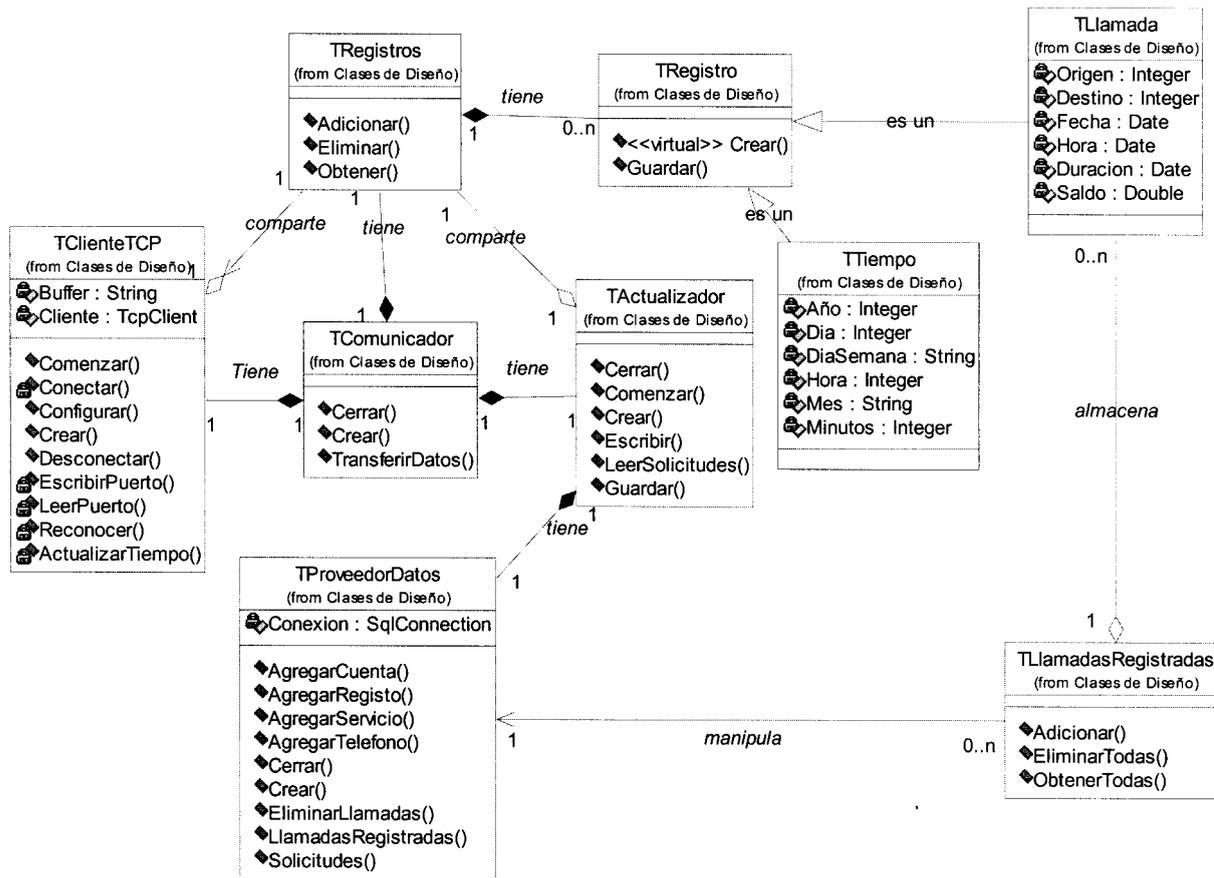
**DCD-CU4: Agregar servicio**



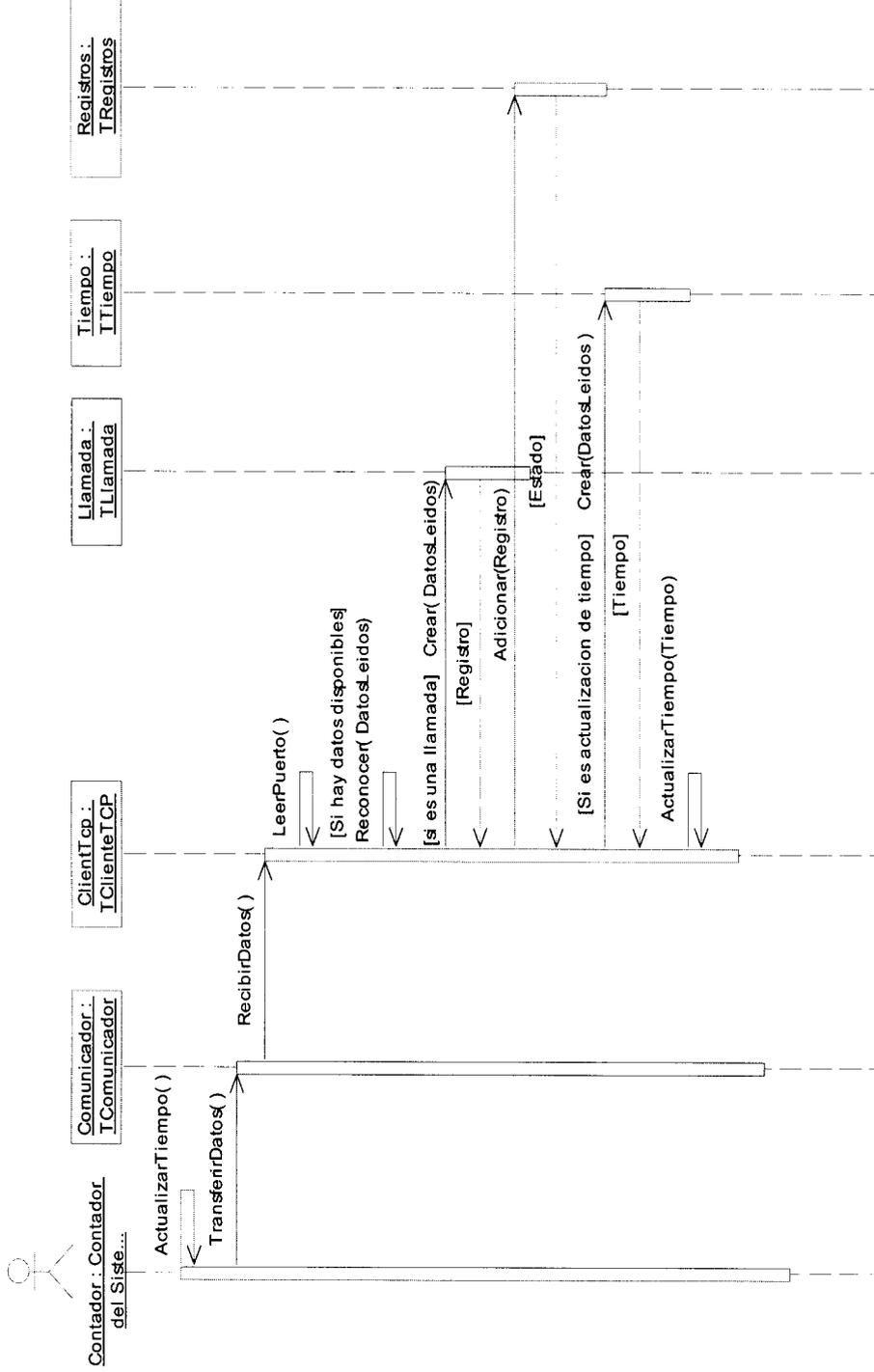
**DI-CU4: Agregar servicio**



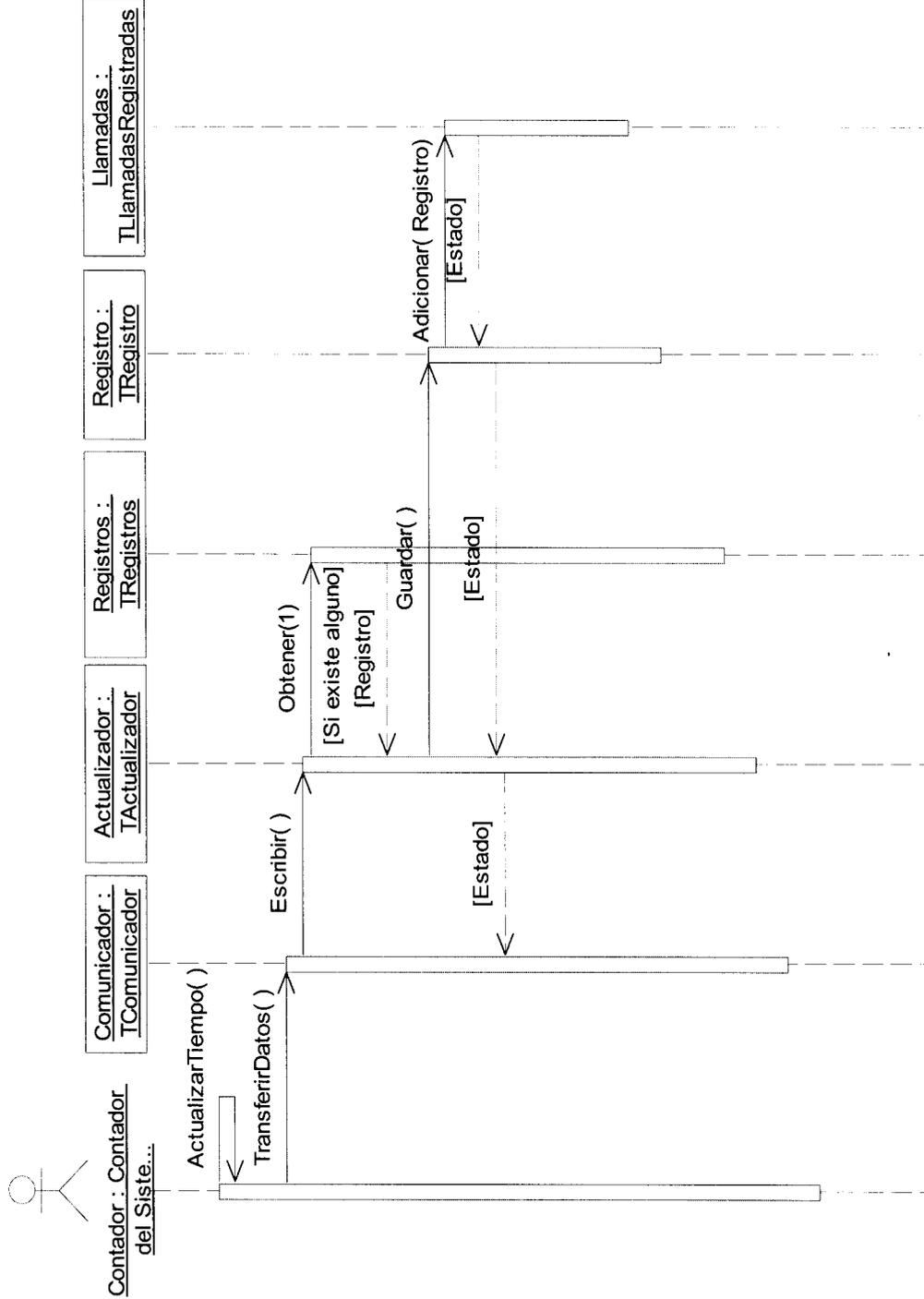
**DCD-CU5: Recepción automática de registros de datos**



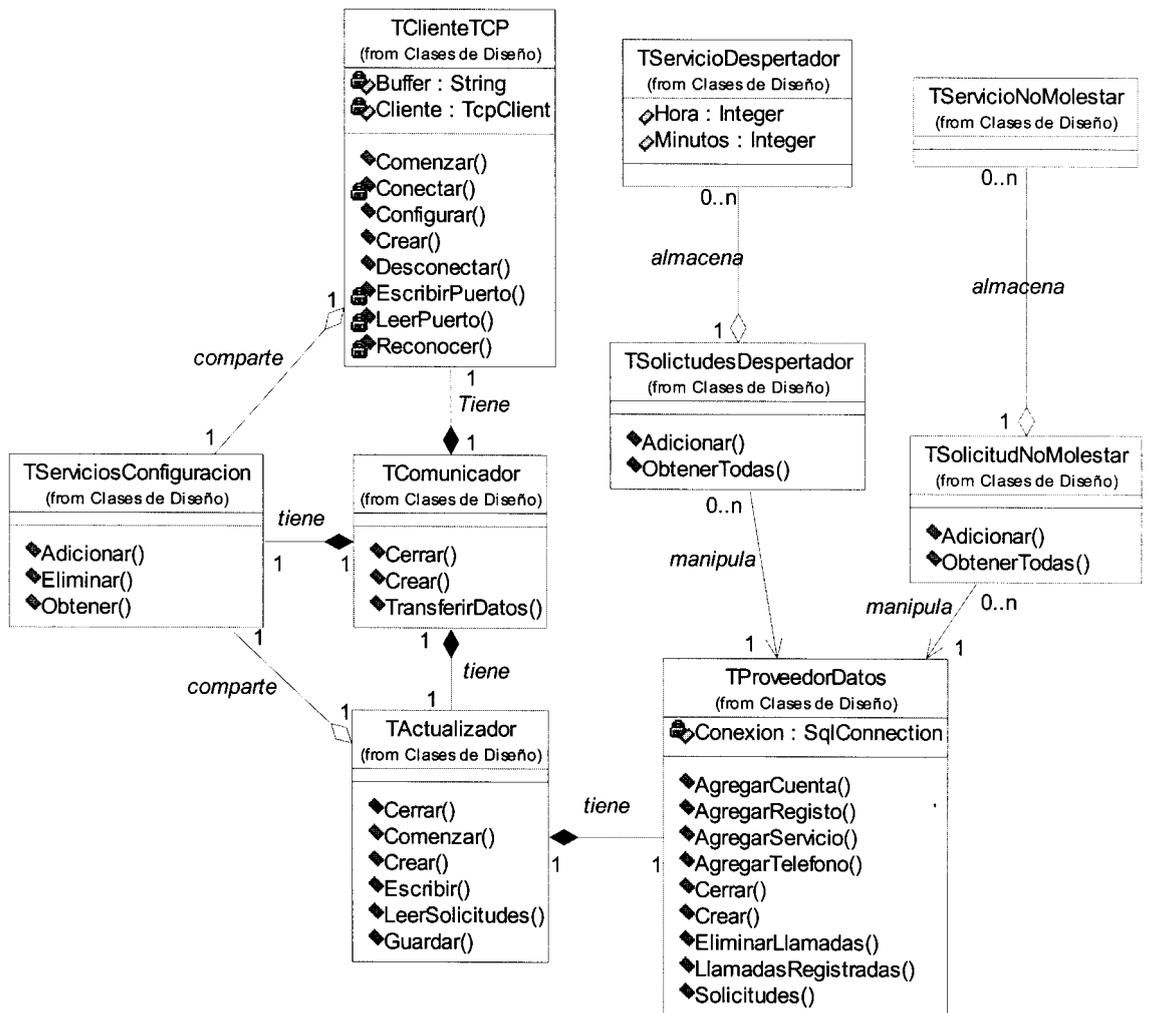
**DI-CU5: Recepción automática de registros de datos (Lectura de TCP/IP)**



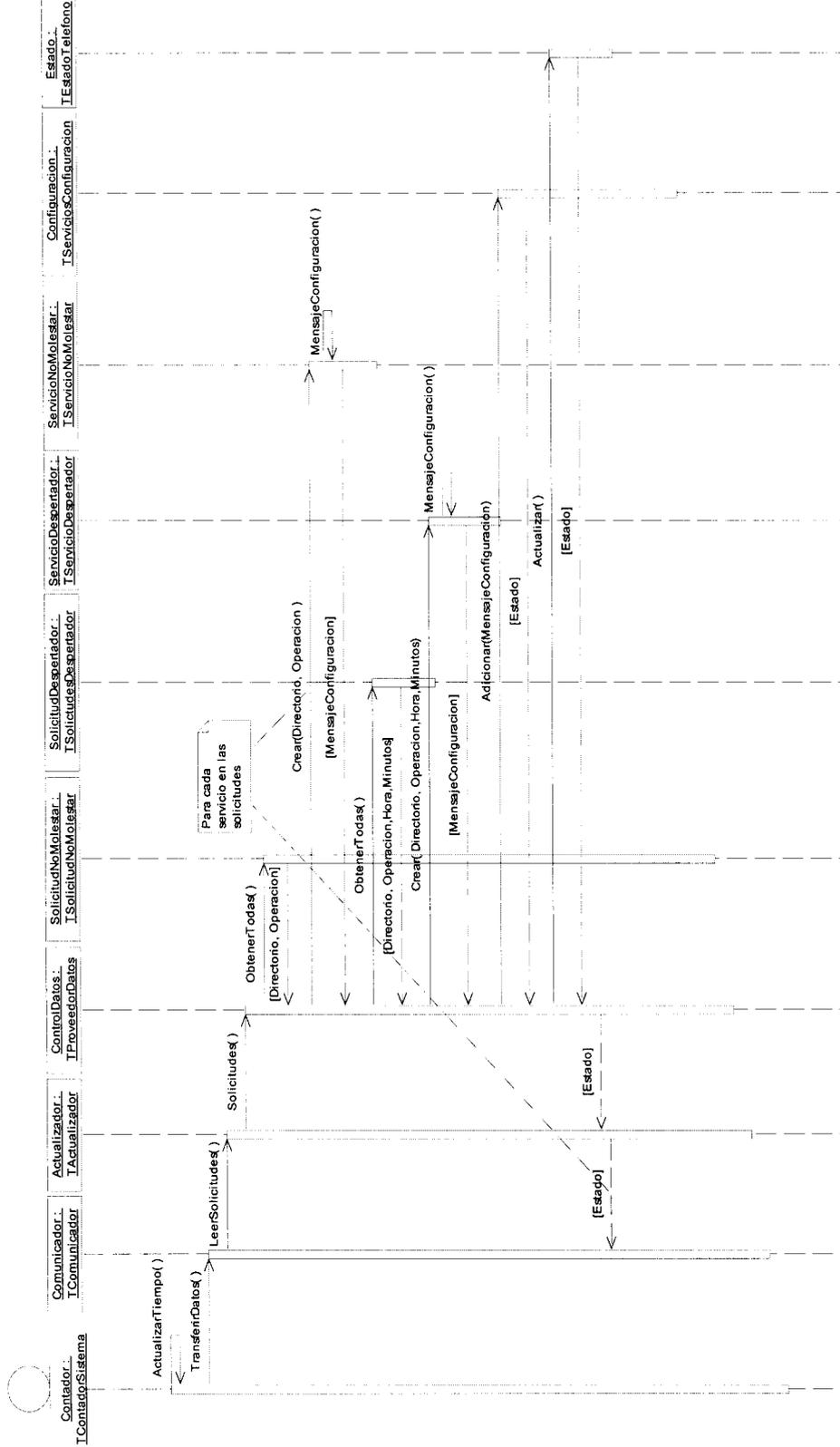
DI-CU5: Recepción automática de registros de datos (escritura en BD)



**DCD-CU6: Configuración automática de servicios**



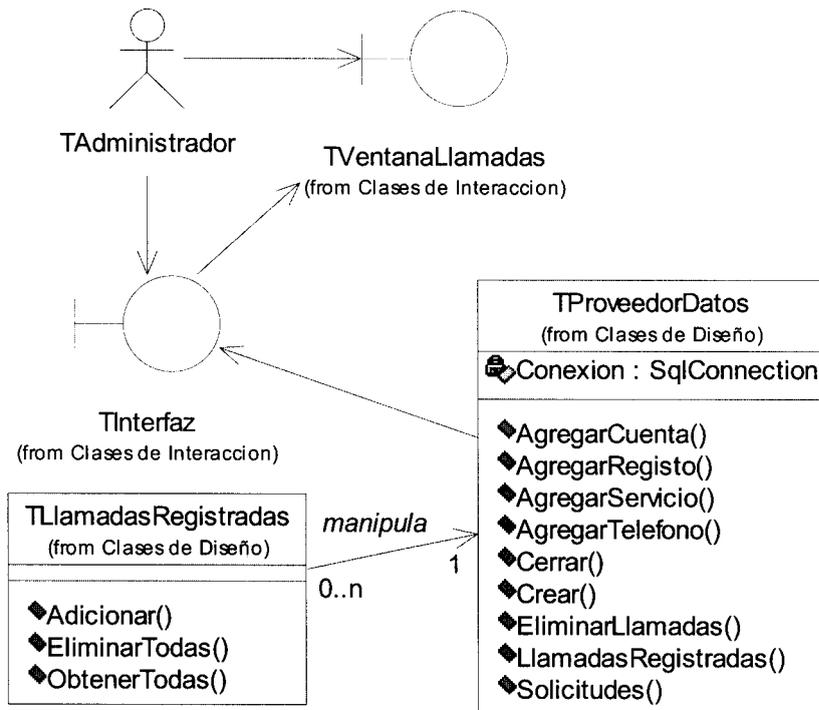
DI-CU6: Configuración automática de servicios (Lectura de BD)



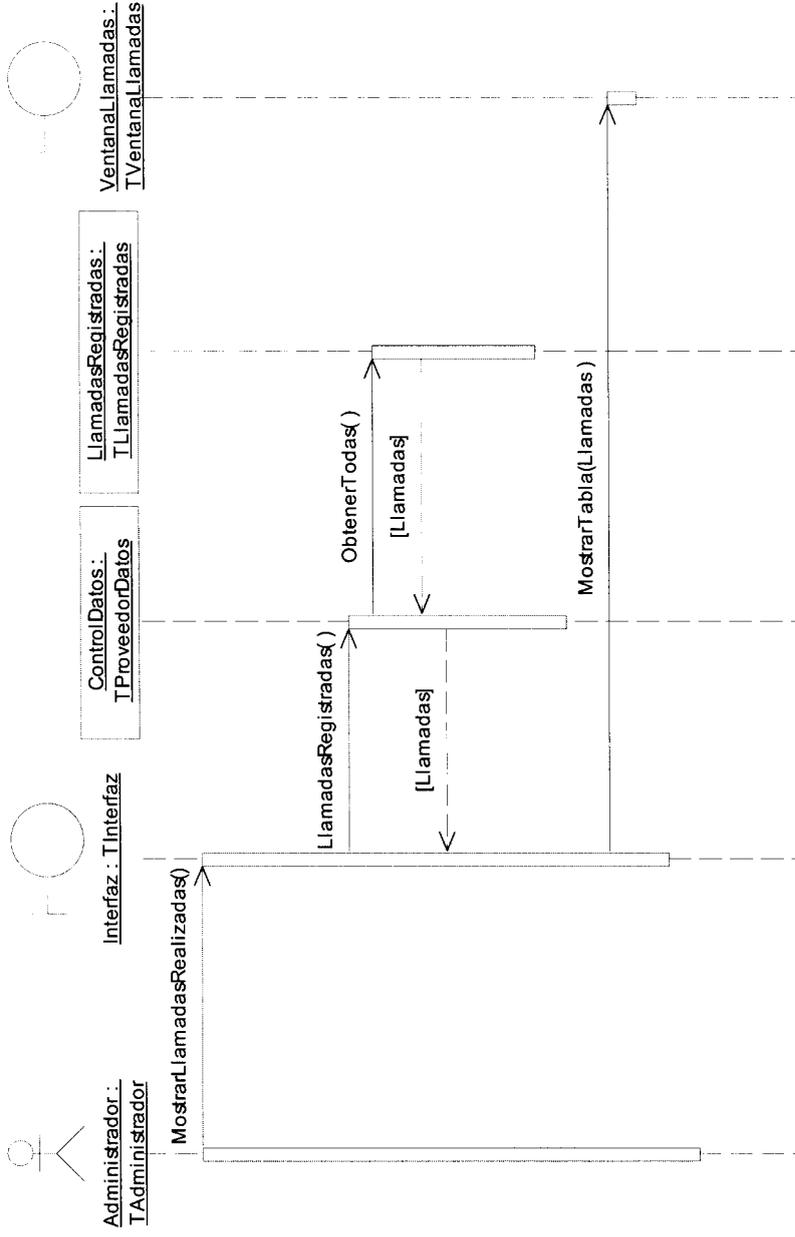
DI-CU6: Configuración automática de servicios (Paquete configuración)



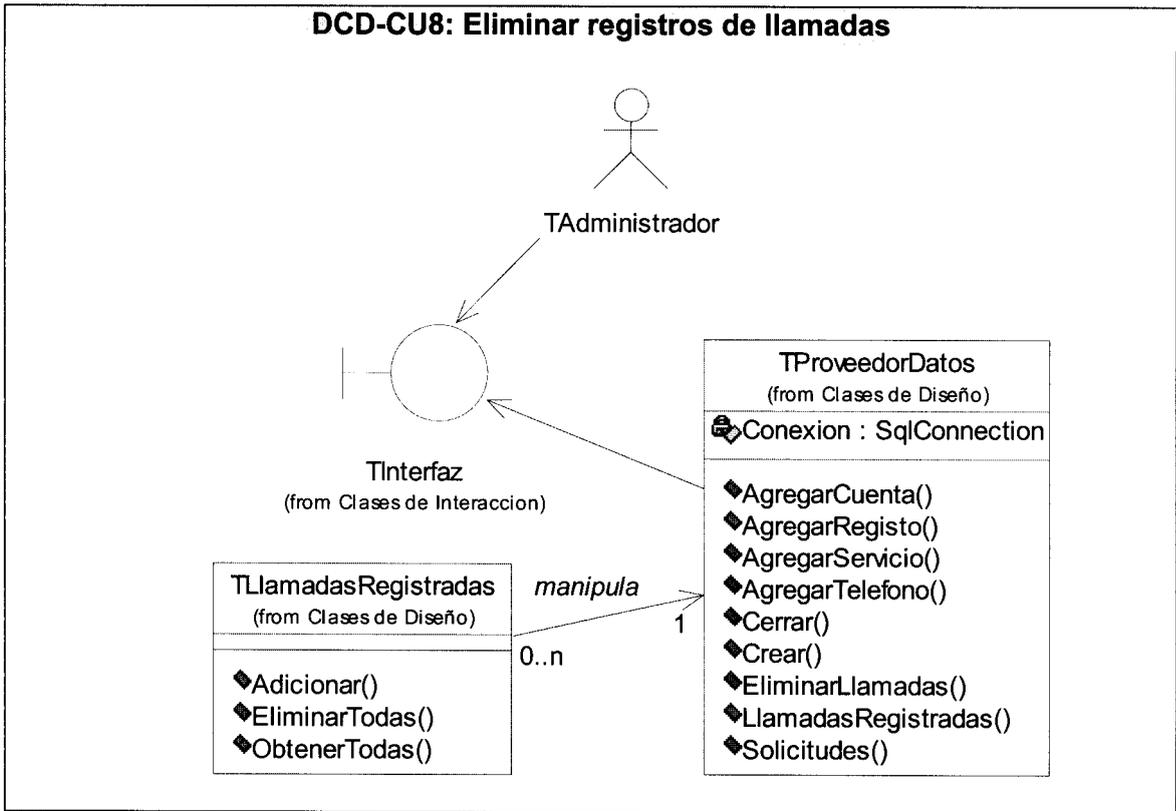
**DCD-CU7: Mostrar llamadas registradas por las líneas del conmutador**



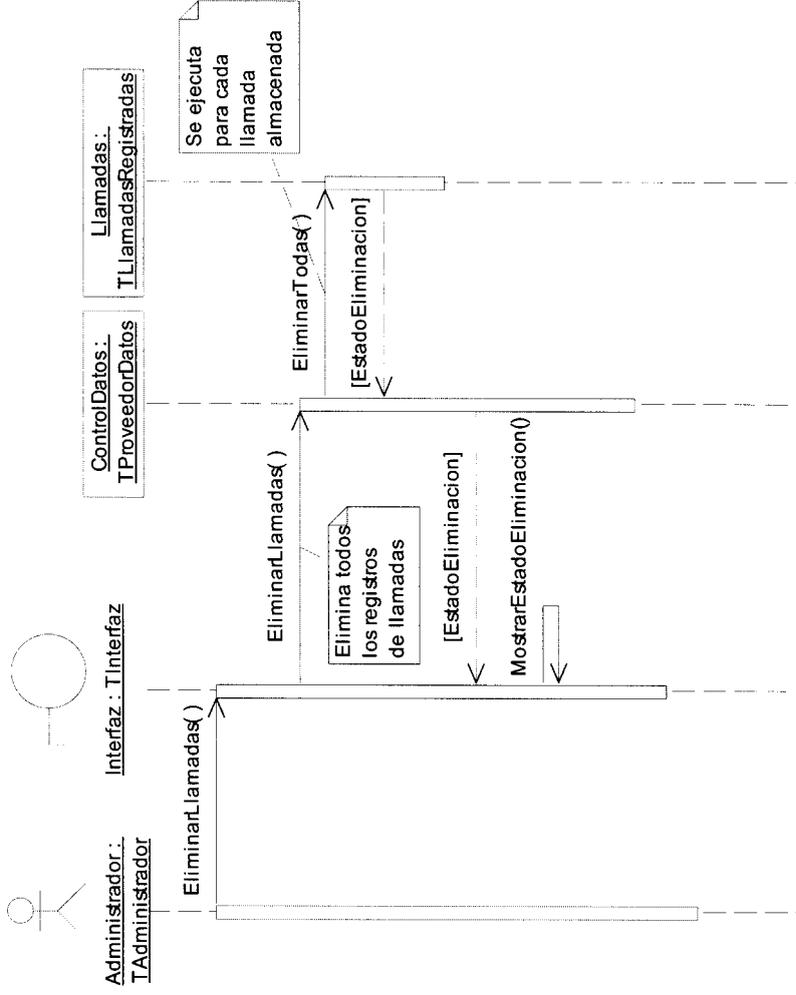
### DI-CU7: Mostrar llamadas registradas por las líneas del conmutador



**DCD-CU8: Eliminar registros de llamadas**



### DI-CU8: Eliminar registros de llamadas



## Anexo 5. Descripción de las clases de diseño.

<b>Nombre:</b> TRegistros	
<b>Para cada responsabilidad:</b>	
Nombre:	Adicionar(Registro: TRegistro)
Descripción:	Adiciona al conjunto de registros almacenado en una instancia de esta clase el elemento pasado como parámetro a este método.
Nombre:	Eliminar(Registros: TRegistro)
Descripción:	Elimina del conjunto de registros almacenados en una instancia de esta clase el primer registro que coincida con las especificaciones del parámetro de este método.
Nombre:	Obtener(posicion: integer): TRegistro
Descripción:	Devuelve del conjunto de registros almacenados en una instancia de esta clase el que se encuentra en la posición indicada como parámetro de este método.

<b>Nombre:</b> TServiciosConfiguracion	
<b>Para cada responsabilidad:</b>	
Nombre:	Adicionar(Servicio: TServicioConfiguracion)
Descripción:	Adiciona al conjunto de servicios almacenado en una instancia de esta clase el elemento pasado como parámetro a este método.
Nombre:	Eliminar(Servicio: TServicioConfiguracion)
Descripción:	Elimina del conjunto de registros almacenados en una instancia de esta clase el primer servicio de configuración que coincida con las especificaciones del parámetro de este método.
Nombre:	Obtener(Posicion: integer): TServicioConfiguracion
Descripción:	Devuelve del conjunto de registros almacenados en una instancia de esta clase el que se encuentra en la posición indicada como parámetro de este método.

<b>Nombre:</b> TComunicador	
<b>Atributo</b>	<b>Tipo</b>
ClienteTcp	TClienteTcp
Actualizador	TActualizador
Lista_Registros	TRegistros
Lista_Servicios	TServiciosConfiguracion
<b>Para cada responsabilidad:</b>	
Nombre:	Crear: TComunicador
Descripción:	Crea una nueva instancia de la clase. Crear se encarga de establecer una conexión con el servidor TCP/IP y con el servidor de datos.
Nombre:	Cerrar: void.
Descripción:	Cierra las conexiones con el servidor TCP/IP y con el servidor de datos.
Nombre:	TrasnferirDatos: void
Descripción:	Se encarga de transportar información entre las partes involucradas en la comunicación establecida por este objeto, o sea, entre el servidor TCP/IP y el servidor de datos.

<b>Nombre:</b> TClienteTcp.	
<b>Atributo</b>	<b>Tipo</b>
Buffer	string
Cliente	TcpClient
ListaRegistros	TRegistros
ListaConfiguraciones	TServiciosConfiguracion
<b>Para cada responsabilidad:</b>	

Nombre:	Crear(Servidor: string, Puerto: Integer, ListaReg: TRegistros, ListaCon: TServicioConfiguracion): TClienteTcp
Descripción:	Crea una nueva instancia de la clase. Crear se encarga de crear un hilo de ejecución que será el encargado de leer desde el puerto indicado en el servidor TCP/IP los datos enviados por el conmutador digital.
Nombre:	Comenzar: void
Descripción:	Inicia el proceso de lectura de datos desde el servidor TCP/IP y de escritura hacia el servidor donde se encuentra la base de datos del sistema.
Nombre:	Conectar(Servidor: string, Puerto: Integer): Boolean
Descripción:	Establece la conexión con el servidor especificado como parámetro en el puerto indicado también como parámetro del método. Devuelve verdadero si se pudo establecer la conexión.
Nombre:	Desconectar: boolean
Descripción:	Cierra la conexión previamente establecida con el servidor TCP/IP. Devuelve verdadero si se pudo completar la operación satisfactoriamente.
Nombre:	LeerPuerto: void
Descripción:	Es el método empleado para obtener los registros enviados por el conmutador digital.
Nombre:	EscribirPuerto(mensaje: byte[])
Descripción:	Envía al conmutador los bytes correspondiente al mensaje de configuración del servicio recibido como parámetro del método.
Nombre:	ActualizarTiempo(Tiempo: TTime): void
Descripción:	Mantiene actualizada en una instancia de esta clase la hora que indica el conmutador.

<b>Nombre:</b> TActualizador.	
<b>Atributo</b>	<b>Tipo</b>
DataBase	TDataBase
ListaReg	TRegistros
ListaConfiguraciones	TServiciosConfiguracion
<b>Para cada responsabilidad:</b>	
Nombre:	Crear(Servidor: string, Puerto: Integer): TActualizador
Descripción:	Crea un a nueva instancia de la clase. Crear se encarga de crear un objeto de tipo TDatabase para acceder y extraer información de la base de datos y crea además dos hilos de ejecución de los cuales uno de ellos se encargara de leer desde la base de datos las solicitudes de servicios de los abonados y el otro se encarga de almacenar los registros provenientes del conmutador.
Nombre:	Comenzar: void
Descripción:	Inicia el proceso de lectura de las solicitudes de servicios desde el servidor de base de datos y el de escritura de los registros contenidos en la lista de registros hacia dicho servidor.
Nombre:	Leer: void
Descripción:	Es el método empleado para la lectura de las solicitudes de servicios desde la base de datos.
Nombre:	Escribir: void.
Descripción:	Es el método empleado para enviar los valores correspondientes a los registros de datos contenidos en la lista de registros hacia el servidor de datos.
Nombre:	Guardar(Registro: TRegistro): Boolean
Descripción:	Método privado que almacena un registro en la base de datos llamando al método guardar de dicho registro.

<b>Nombre:</b> TTelefono.	
<b>Atributo</b>	<b>Tipo</b>
NumeroDirectorio	integer
Cuota	string
TipoCuota	string
<b>Para cada responsabilidad:</b>	
Nombre:	Crear: TTelefono
Descripción:	Crea una instancia de la clase.

<b>Nombre:</b> TTelefonos.	
<b>Para cada responsabilidad:</b>	
Nombre:	Adicionar(directorio: string, tipocuenta: string: local: integer)
Descripción:	Agregar los datos de un teléfono al conjunto de datos almacenado en este objeto.

<b>Nombre:</b> TServicioConfiguracion (clase abstracta).	
<b>Atributo</b>	<b>Tipo</b>
NumeroDirectorio	integer
Operacion	boolean
<b>Para cada responsabilidad:</b>	
Nombre:	Crear(directorio: integer, operación: boolean): TServicioConfiguracion
Descripción:	Crea una nueva instancia de la clase. Inicializa los atributos de una instancia de esta clase con los parámetros indicado.
Nombre:	MensajeConfiguracion: void
Descripción:	Método abstracto empleado para la configuración del mensaje de habilitación o cancelación de un servicio en el conmutador.

<b>Nombre:</b> TServicioDespertador (hereda de Servicio).	
<b>Atributo</b>	<b>Tipo</b>
Hora	integer
Minutos	integer
<b>Para cada responsabilidad:</b>	
Nombre:	Crear(directorio: integer, operación: boolean, hora: integer, minutos: integer): TServicioDespertador
Descripción	Crea una nueva instancia de la clase. Inicializa los atributos de la misma con los parámetros indicado.
Nombre:	MensajeConfiguracion: void
Descripción:	Método empleado para obtener el mensaje de habilitación o cancelación del servicio de despertador en el conmutador mediante los atributos contenido en una instancia de esta clase.

<b>Nombre:</b> TSolicitudDespertador.	
<b>Para cada responsabilidad:</b>	
Nombre:	Adicionar(directorio: integer, operación: boolean, hora: integer, minutos: integer)
Descripción:	Agregar los datos de una solicitud de este servicio al conjunto de datos almacenado en este objeto.
Nombre:	ObtenerTodas: DataTable
Descripción:	Devuelve todas las solicitudes del servicio de despertador almacenados en una instancia de esta clase.

<b>Nombre:</b> TServicioNoMolestar (hereda de Servicio).	
<b>Para cada responsabilidad:</b>	
Nombre:	Crear(directorio: integer, operación: boolean): TServicioNoMolestar
Descripción:	Crea una nueva instancia de la clase. Inicializa los atributos de una instancia de esta clase con los parámetros indicados.
Nombre:	MensajeConfiguracion: void
Descripción:	Método empleado para obtener el mensaje de habilitación o cancelación del servicio no molestar en el conmutador mediante los atributos de una instancia de esta clase.

<b>Nombre:</b> TSolicitudNoMolestar.	
<b>Para cada responsabilidad:</b>	
Nombre:	Adicionar(directorio: integer, operación: boolean)
Descripción:	Agregar los datos de una solicitud este servicio al conjunto de datos almacenado en este objeto.
Nombre:	ObtenerTodas: DataTable
Descripción:	Devuelve todas las solicitudes del servicio de no molestar almacenados en una instancia de esta clase.

<b>Nombre:</b> TEstadoTelefono	
<b>Para cada responsabilidad:</b>	
Nombre:	Actualizar(directorio: integer, hora: integer, minutos: integer, nomolestar: boolean)
Descripción:	Actualiza en la lista de objetos contenidos en una instancia de esta clase los datos correspondientes a los servicios habilitados o no para el directorio especificado.

<b>Nombre:</b> TRegistro (clase abstracta)	
<b>Para cada responsabilidad:</b>	
Nombre:	Crear: TRegistro
Descripción:	Crea un a nueva instancia de la clase. Es un objeto abstracto por lo que solo se utiliza para representar los distintos tipos de registros de mensajes que puede enviar el conmutador.
Nombre:	Guardar(): boolean
Descripción:	Se encarga de almacenar el registro en la base de datos en dependencia del tipo de registro. Debe ser redefinido en cada uno de los registros herederos

<b>Nombre:</b> TLLamada (hereda de Registro)	
<b>Atributo</b>	<b>Tipo</b>
Origen	integer
Destino	integer
Hora	date
Duracion	date
Fecha	date
Saldo	double
<b>Para cada responsabilidad:</b>	
Nombre:	Crear(origen: integer, destino: integer, hora: date, duracion: date, fecha: date, saldo: double): TLLamada
Descripción:	Crea un a nueva instancia de la clase. Inicializa los atributos de una instancia de esta clase con los parámetros establecidos en este método.
Nombre:	Guardar(): boolean
Descripción:	Se encarga de almacenar el registro de llamada en la correspondiente tabla de la base de datos habilitada para ello.

<b>Nombre:</b> TTiempo(hereda de Registro)	
<b>Atributo</b>	<b>Tipo</b>
Hora	integer
Minutos	integer
Día	integer
Mes	string
Año	integer
DiaSemana	string
<b>Para cada responsabilidad:</b>	
Nombre:	Crear(hora: integer, min: integer, dia: integer, mes: string, año: integer, diasemana: string): TTiempo
Descripción:	Crea un a nueva instancia de la clase. Inicializa los atributos de una instancia de esta clase con los parámetros establecidos en este método.
Nombre:	Guardar(): boolean
Descripción:	Se encarga de almacenar el registro de tiempo en la correspondiente tabla de la base de datos habilitada para ello.

<b>Nombre:</b> TServicios.	
<b>Para cada responsabilidad:</b>	
Nombre:	Adicionar(nombre: string, tipo: string, descripción: string)
Descripción:	Agregar los datos de un servicio al conjunto de datos almacenado en este objeto.
Nombre:	ObtenerTodos: DataTable
Descripción:	Devuelve los datos de todos los servicios almacenados en este objeto.

<b>Nombre:</b> TTiposCuenta.	
<b>Para cada responsabilidad:</b>	
Nombre:	Adicionar(tipo: string, descripción: string, cantidad: double)
Descripción:	Agregar los datos de un tipo de cuenta al conjunto de datos almacenado en este objeto.
Nombre:	ObtenerTodos: DataTable
Descripción:	Devuelve los datos de todos los servicios almacenados en este objeto.

<b>Nombre:</b> TTiempoRegistrado.	
<b>Para cada responsabilidad:</b>	
Nombre:	Adicionar(tiempo: date)
Descripción:	Agregar los datos tiempo pasado como parámetro a este tipo de objeto.
Nombre:	Obtener: DataTable
Descripción:	Devuelve los datos de tiempo almacenado en este objeto.

<b>Nombre:</b> TProveedorDatos	
<b>Atributo</b>	<b>Tipo</b>
Conexión	SqlConnection
<b>Para cada responsabilidad:</b>	
Nombre:	Crear(Servidor: string, DB: string)
Descripción:	Crea un a nueva instancia de la clase. Crear se encarga de crear un objeto de conexión con la base de datos.
Nombre:	Cerrar: boolean
Descripción:	Cierra la conexión establecida con la base de datos que se encuentra almacenada en el objeto de conexión de esta clase.

Nombre:	AragarTelefono(directorio: string, tipocuenta: string, local: integer)
Descripción:	Adiciona a la base de datos indicada en el objeto conexión los datos de un teléfono contenidos en los parámetros de este método.
Nombre:	AgregarCuenta(tipo: string, descripción: string, cantidad: double)
Descripción:	Agrega un nuevo tipo de cuenta a la base de datos indicada en el objeto de conexión de esta clase.
Nombre:	AgregarServicio(Nombre: string, Tipo: string, Descripción: string)
Descripción:	Agrega un nuevo servicio a la base de datos indicada en el objeto de conexión de esta clase.
Nombre:	EliminarRegistrosLlamadas: boolean
Descripción:	Elimina de la base de datos indicada en el objeto de conexión de esta clase los registros correspondientes a las llamadas registradas por el conmutador digital. Devuelve el estado de la eliminación.
Nombre:	LlamadasRegistradas: DataTable
Descripción:	Obtiene de la base de datos indicada en el objeto de conexión de esta clase las llamadas almacenadas que han sido registradas por el conmutador digital.
Nombre:	Solicitudes: DataTable
Descripción:	Método para obtener de la base de datos indicada en el objeto de conexión de esta clase las solicitudes de configuración de servicios telefónicos.

## Anexo 6. Descripción de las tablas de la base de datos.

<b>Nombre: Telefono</b>		
<b>Descripción:</b> Contiene los números de los directorios que participan en las llamadas telefónicas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
NoTelefono	varchar	Número de directorio. Es la llave de la tabla.

<b>Nombre: TelefonoUCI</b>		
<b>Descripción:</b> Contiene información relacionada con los teléfonos locales.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
NoTelefono	varchar	Número de directorio. Es la llave de la tabla.
IdLocal	int	Identificador del local donde se encuentra ubicado el teléfono.
IdCuenta	tinyint	Identificador del tipo de cuenta asociado al directorio.
Hora	char	Indica la hora para la que está configurado el servicio de despertador automático.
Minutos	char	Indica los minutos para los está configurado el servicio de despertador automático.
No_Molestar	bit	Indica el estado del servicio de no molestar (habilitado o cancelado).
Colgado	bit	Indica si el teléfono está colgado o descolgado.

<b>Nombre: Localidad</b>		
<b>Descripción:</b> Contiene información acerca de los códigos de teleselección para las llamadas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idTeleseleccion	smallint	Identificador del código de teleselección. Es la llave de la tabla.
codigo	char	Dígitos de la teleselección.
nombre	nvarchar	Localización con este código de teleselección.
idProvincia	tinyint	Identificador de la provincia a la que pertenece la localidad.

<b>Nombre: Provincia</b>		
<b>Descripción:</b> Contiene las provincias del país.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idProvincia	tinyint	Identificador de la provincia. Es la llave de la tabla.
nombre	nvarchar	Nombre de la provincia.

<b>Nombre: Llamada</b>		
<b>Descripción:</b> Contiene información relacionada con las llamadas telefónicas registradas por el conmutador Infinity.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
NoTelefono	varchar	Número del directorio que origina la llamada. Constituye parte de llave de la tabla.
NoDestino	varchar	Número del directorio destino de la llamada. Contiene todos los dígitos marcados en el origen.
IdTeleseleccion	smallint	Identificación del lugar de destino de la llamada. Viene dado por el código de teleselección.
fechaLlamada	datetime	Fecha de realización de la llamada. Es parte de llave de la tabla.
duracion	char	Duración de la llamada.
saldo	float	Costo generado por la llamada.

<b>Nombre: Cuenta</b>		
<b>Descripción:</b> Contiene información de las cuentas telefónicas		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idCuenta	tinyint	Identificador de la cuenta. Es la llave de la tabla.
idTipoCuenta	tinyint	Identificador del tipo de cuenta.
cuota	float	Cuota inicial asignada a la cuenta.

<b>Nombre: TipoCuenta</b>		
<b>Descripción:</b> Contiene los tipos de cuentas telefónicas los teléfonos		
Atributo	Tipo	Descripción
idTipoCuenta	tinyint	Identificador de tipo de cuenta. Es la llave de la tabla.
tipo	varchar	Nombre del tipo de cuenta.
descripcion	varchar	Descripción del tipo de cuenta.

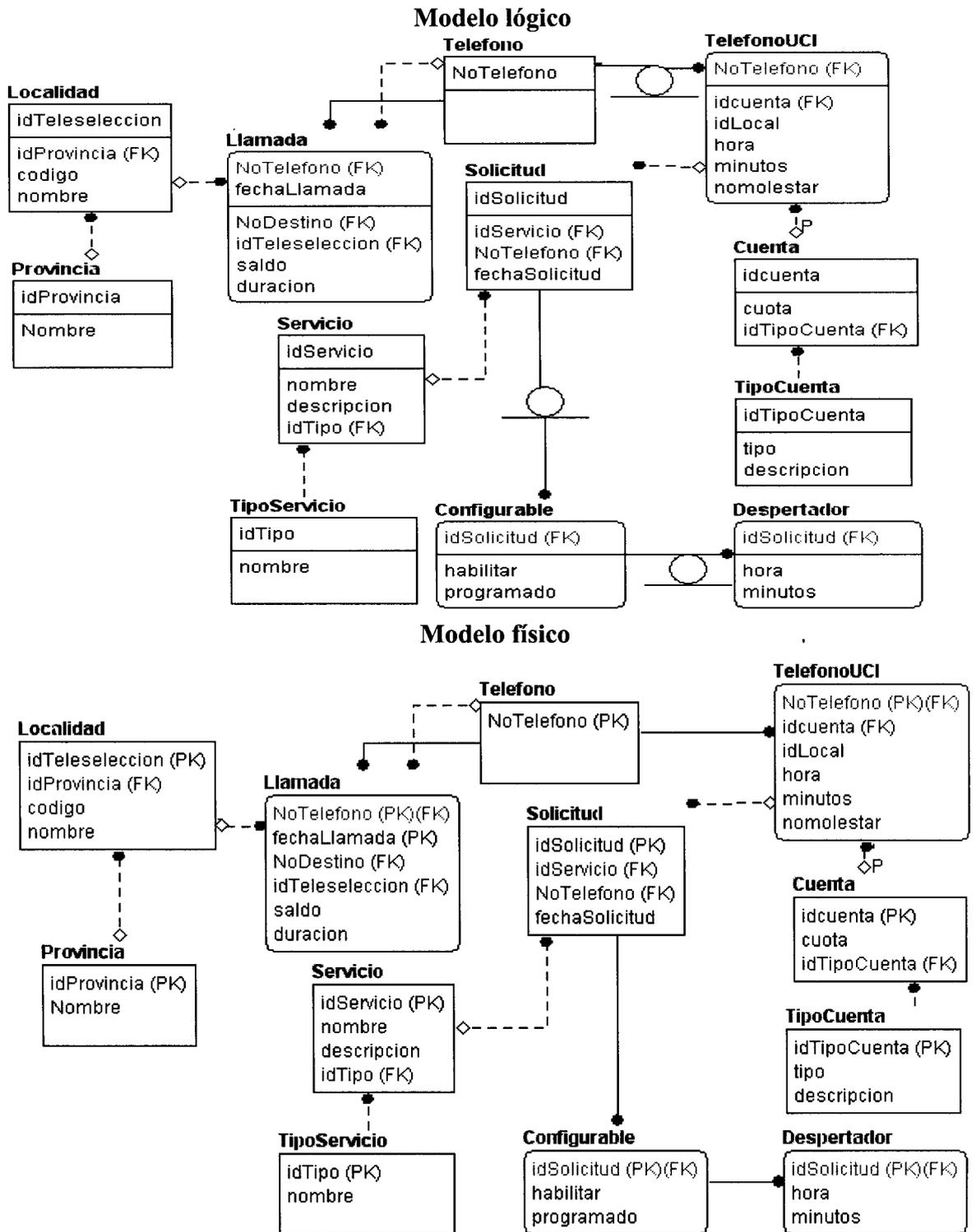
<b>Nombre: Solicitud</b>		
<b>Descripción:</b> Contiene información sobre los servicios telefónicos.		
Atributo	Tipo	Descripción
IdSolicitud	int	Identificar de la solicitud. Es la llave de la tabla.
NoTelefono	char	Número del directorio que solicita el servicio.
idServicio	tinyint	Identificador del tipo de servicio.
fechaSolicitud	datetime	Fecha de realización de la solicitud.

<b>Nombre: Servicio</b>		
<b>Descripción:</b> Contiene información sobre los servicios telefónicos.		
Atributo	Tipo	Descripción
idServicio	tinyint	Identificador del servicio telefónico. Es la llave de la tabla.
nombre	varchar	Nombre del servicio telefónico.
tipo	int	Tipo de servicio telefónico.
descripción	varchar	Breve descripción del servicio telefónico.

<b>Nombre: Configurable</b>		
<b>Descripción:</b> Almacena los datos para la configuración del servicio no molestar de los teléfonos.		
Atributo	Tipo	Descripción
idSolicitud	int	Identificador de la solicitud de este servicio. Es la llave de la tabla.
NoTelefono	char	Número del directorio que solicita el servicio.
idServicio	tinyint	Identificador del servicio.
habilitar	bit	Acción a realizar (habilitar o cancelar)
programado	bit	Determina si la solicitud fue atendida o no.

<b>Nombre: Despertador</b>		
<b>Descripción:</b> Almacena los datos para la configuración del servicio de despertador automático de los teléfonos.		
Atributo	Tipo	Descripción
idSolicitud	int	Identificador de la solicitud de este servicio. Es la llave de la tabla.
NoTelefono	char	Número del directorio que solicita el servicio.
idServicio	tinyint	Identificador del servicio.
Hora	int	Hora para la configuración del servicio.
Minutos	Int	Minutos para la configuración del servicio.

## Anexo 7. Diseño lógico y físico de la base de datos.



## **Glosario de términos.**

---

**Abonado:** Persona que recibe algún servicio periódicamente o determinado número de veces.

**Central digital:** Central que conmuta información en forma digital a través de sus órganos de conmutación.

**Conmutación:** Establecimiento, a petición, de una conexión individual entre una entrada y una salida deseadas dentro de un conjunto de entradas y salidas durante el tiempo necesario para la transferencia de la información.

**Multiplexor:** Circuito digital que combina varias señales de entradas en una única salida conmutando entre ellas a una frecuencia constante.

**Protocolo:** Declaración formal de los procedimientos que se adoptan para realizar la comunicación entre dos o más funciones dentro de una misma capa de una jerarquía de funciones.

**Registro:** Conjunto de datos o palabras tratados como una unidad.

**Sistema automático:** Sistema en el que las operaciones de conmutación se efectúan por medio de aparatos controlados eléctricamente, sin intervención de operadora.